APPLICATION OF A STATE-SPACE WAKE MODEL TO A SERVO FLAP CONTROLLED ROTOR IN HOVER

A THESIS

presented to

The Academic Faculty

by

Martin Stettner

In Partial Fulfillment

of the Requirements for the Degree

Master of Science

School of Aerospace Engineering

Georgia Institute of Technology

December 1990

APPLICATION OF A STATE-SPACE WAKE MODEL TO A SERVO FLAP CONTROLLED ROTOR IN HOVER

Approved:

David A. Peters, Chairman

Narayanamoorthy L. Sankar

Chien-Hsiung Chuang

Date approved by Chairman: Dec 11, 1990

Acknowledgements

I would like to express deep felt gratitude to Dr. David A. Peters, my thesis advisor, for his invaluable help, guidance and encouragement throughout this thesis preparation. His never ending patience, exceptional scholar spirit, and friendship with his students will always be a great example to me.

I also gratefully acknowledge Prof. Narayanamoorthy L. Sankar and Prof. Chien-Hsiung Chuang, the members of the reading committee, for their careful reading and their suggestions.

Finally, I owe special debt to my family for their continuous support, and to that small group of very special friends here in Atlanta for their comradeship and encouragement. Their friendship was and is essential to me.

Table of Contents

List of Tables

List of Figures

List of Symbols

A	normalwash at blade feathering axis, [ft/sec]
A_{nj}^{m}	matrix of integrals; closed form solution in (4.20)
a	two-dimensional lift curve slope, rotor blade without flap
B	effective blade pitch rate, [rad/sec]
B_{nj}^{m}	matrix of integrals; closed form solution in (4.21)
$[CXX]$	damping matrix (matrix notation see section 5.1)
c	blade mean chord, $\bar{c}\,R$ [ft]
c_f	servo flap chord, $\bar{c}_f\,R$ [ft]
c_l	lift coefficient
c_{l0}	lift coefficient at zero effective angle of attack
$c_{l\delta}$	derivative of lift coefficient with respect to flap deflection
c_m	moment coefficient
c_{m0}	moment coefficient at zero effective angle of attack
$c_{m\delta}$	derivative of moment coeff. with respect to flap deflection
$\{cX\}$	constant vector (matrix notation see section 5.1)
e	flapping hinge offset, $\bar{e}\,R$ [ft]
e_A	offset blade feathering axis - aerodynamic center, $\bar{e}_A\,\bar{c}\,R$ [ft]
e_f	offset blade feathering axis - flap hinge line, $\bar{e}_{fc}\,\bar{c}\,R$ [ft]
H_n^m	combination of factorials, eq. (4.19)
$I_{\beta\beta}$	flapping moment of inertia, [slugs-ft^2]
$I_{\beta\theta}$	flapping - feathering coupling moment of inertia, [slugs-ft^2]
$I_{\beta\delta}$	flapping - flap defl. coupling moment of inertia, [slugs-ft^2]
$I_{\delta\delta}$	flap deflection moment of inertia, [slugs-ft^2]

$I_{\theta\theta}$	feathering moment of inertia, [slugs-ft^2]
$\bar{I}_{\beta\delta}, \bar{I}_{\beta\theta}, \bar{I}_{\delta\delta}, \bar{I}_{\theta\theta}$	nondimensional elements of blade dynamics inertia matrix, normalized by $I_{\beta\beta}$
i	$=\sqrt{-1}$
K_β	hub spring rate, flapping , $\bar{K}_\beta \Omega^2 I_{\beta\beta}$ [ft-lb/rad]
K_c	hub spring rate, coupling, $\bar{K}_c \Omega^2 I_{\beta\beta}$ [ft-lb/rad]
K_θ	hub spring rate, feathering, $\bar{K}_\theta \Omega^2 I_{\beta\beta}$ [ft-lb/rad]
$\bar{K}_{\beta\beta}, \bar{K}_{\beta\delta}, \bar{K}_{\beta\theta}$, $\bar{K}_{\delta\delta}, \bar{K}_{\theta\theta}$	nondimensional elements of balde dynamics stiffness matrix, normalized by $I_{\beta\beta}$
K_n^m	apparent mass diagonals, eq. (4.18)
[KXX]	stiffness matrix (matrix notation see section 5.1)
$\bar{L}$	sectional lift, normalized by $\rho \Omega^2 R^3$
$\bar{M}$	sectional moment, normalized by $\rho \Omega^2 R^4$
M	mass, [slugs]; also highest harmonic ocurring in inflow azimuthal expansion function
M^A	aerodynamic moment, [ft-lb]
$\bar{M}_\delta$	control and aerodynamic moments about servo flap hinge
[MXX]	mass matrix (matrix notation see section 5.1)
m	harmonic number
N	highest polynomial number occurring in inflow expressions
n	polynomial number
P_n^m	Legendre functions of first kind
$\bar{P}_n^m$	normalized Legendre functions (eq. (4.8))
Q	number of blades
q	perturbation velocity, also blade index

R	rotor radius, [ft]
{RHS}	right hand side vector ([R] {u̇})
r	radial coordinate, $\bar{r}$ R [ft]
{r}	position vector to point on the rotor, [ft]
S	number of shape functions
S_β	static moment about flapping hinge, $\bar{S}_\beta$ ($I_{\beta\beta}$ / R), [slugs-ft]
S_θ	static moment about feathering axis, $\bar{S}_\theta$ ($I_{\beta\beta}$ / R), [slugs-ft]
S_δ	static moment about feathering axis, $\bar{S}_\delta$ ($I_{\beta\beta}$ / R), [slugs-ft]
t	time, $\bar{t}$ Ω, [sec]
T	kinetic energy
[T1]	transformation matrix, includes sin and cos terms of the $\tilde{\tau}_n^{mc}$ and $\tilde{\tau}_n^{ms}$
[T2]	transformation matrix, includes sin and cos terms of the azimuthal inflow expansion
{u}	vector of control variables and their time derivatives
V	potential energy
{v}	velocity vector of a point on the rotor, [ft/sec]
w	normalwash, $\bar{w}$ ΩR [ft/sec]
[X]	modal matrix
x	chordwise coordinate on blade, $\bar{x}$ R [ft]
x^f	chordwise coordinate on servo flap, [ft]
{x}	vector of blade state variables
{y}	vector of inflow state variables
z_0	vertical coordinate of blade feathering axis or flap hinge line, $\bar{z}_0$ R [ft]

Greek Symbols:

α_n^m	induced inflow states (cos-partition), non-rotating system
β	flapping angle, [rad]
β_n^m	induced inflow states (sin-partition), non-rotating system
$\bar{\beta}$	precone, [rad]
γ	Lock number
δ	servo flap deflection angle, [rad]
δ_c	servo flap control input, [rad]
δ_{ij}	Kronecker delta; 1 for i = j, 0 otherwise
ε	reference unit for ordering schemes $[10^{-1}]$
η	feedback ratio flapping - servo flap deflection; also ellipsoidal coordinate
κ^{bf}	ratio lift curve slopes flapped - unflapped blade section
κ^f	ratio lift curve slopes servo flap - unflapped blade section
λ	rotor inflow, normalized by rotor tip speed ΩR
μ	advance ratio
μ^b, μ^f	structural density of blade and flap, [slugs/ft^3]
ν	ellipsoidal coordinate on the rotor disc $\sqrt{1 - \bar{r}^2}$
ρ	density of air, [slugs/ft^3]
Φ	perturbation pressure
ϕ	pressure on the rotor disk
φ_q	phase of control input blade q wrt. reference blade
ψ	azimuthal coordinate, [rad]
$\bar{\psi}$	ellipsiodal coordinate, on the rotor disk $\bar{\psi} = \psi$
$\tilde{\psi}$	azimuthal position, rotating frame, [rad]
$\tilde{\psi}_q$	position of blade q with respect to a reference blade

	in the rotating system, [rad]
τ_n^m	inflow dynamics forcing function
θ	feathering angle, [rad]
$\bar{\theta}$	feathering angle ref. value (retention strap pretwist), [rad]
ζ	feedback ratio feathering - servo flap deflection
ζ^f	ratio servo flap chord - blade chord
χ	wake skew angle, [rad]
X	$= \tan\lvert\chi/2\rvert$
ξ	freestream direction
Ω	rotor speed, [rad/sec]
ω	natural frequency, $\bar{\omega}\,\Omega$, [rad/sec]
ω_R	frequency, rotating system, $\bar{\omega}_R\Omega$, [rad/sec]

Operators:

$(\,)^*$	derivative with respect to nondimensional time, $(\dot{}) / \Omega$
$\dot{(\,)}$	time derivative
$\overline{(\,)}$	nondimensional value
$\widetilde{(\,)}$	amplitude
$\{(\,)\}$	vector
$\{(\,)\}_j$	vector of column j of a matrix
$[(\,)]$	matrix
C(( ))	dynamic inflow operator
L(( ))	static inflow operator
O[( )]	order of ( ) / of order ( )

Subscripts:

$()_{nc}$	non-circulatory (apparent mass term)
$()_q$	blade q
$()_{qs}$	quasi-steady
$()_{rad}$	radial flow contribution
$()_\beta$	about flapping hinge
$()_\delta$	about flap hinge line
$()_\theta$	about feathering axis

Superscripts:

$()^b$	blade
$()^{bf}$	blade and flap, rotor blade flapped section
$()^c$	cos-partition (inflow dynamics)
$()^f$	servo flap
$()^s$	sin-partition (inflow dynamics)
$()^I$	system I
$()^{II}$	system II
$()^{III}$	system III

SUMMARY

An investigation is made of the aeroelastic behavior of a servo flap controlled rotor in hover with the intention of providing a tool for future research on the dynamic behavior of this configuration. Quasi-steady aerodynamics and apparent mass terms based on thin airfoil theory are correlated by test data. Rotor dynamics are then represented by coupled flapping and feathering of rigid blades, accounting for mechanical feedback as utilized at the KAMAN 101 Main Rotor [14]. Induced flow is found from the dynamic inflow model of Peters and He [9-11], which implies unsteady aerodynamics.

In order to provide the bases for extensions, the development of the components blade aerodynamics and rotor dynamics, and the application of Peters' state-space wake model are discussed in detail. The characteristics of the rotor dynamics are correlated with results from Kaman Aerospace [13]. The coupled rotor-wake system has been implemented in a FORTRAN code. This code includes the non-linear wake model for trim calculations in hover and a linear version has been utilized for obtaining the system's response to collective, progressing, regressing, and differential higher harmonic control (HHC) input modes. Additionally, the rotor response to deflections of one servo flap only can be investigated as a preparation for individual blade control (IBC) applications.

Results show that the deflected servo flap requires significantly more radial expansion functions for the trim states to converge; even for a maximum polynomial order of N = 12 satisfactory convergence does not occur.

On the other hand, the dynamic response to higher harmonic control inputs at 3.0/rev, 4.0/rev, and 5.0/rev can be considered as converged for N = 8, supporting the findings of He[11] and Su[12] for conventional rotors. The overall rotor response at these frequencies is small; still, the effect of the characteristic wake modes can be detected. Finally, the model captures the complex interactions between the rotor blades for the individual blade actuation. The flapping response of the non-actuated blades drops sharply around the flapping natural frequency, and the amplitudes are larger the closer the observed blade follows the actuated blade. In both HHC and IBC application the wake influence on the feathering response is small, if not negligible.

CHAPTER 1

INTRODUCTION

1.1 Servo Flap Control System in HHC

In forward flight the blades of a helicopter move through a constantly changing aerodynamic environment resulting from the addition of forward flight speed, rotor rotation, and the disturbances generated by the preceding blades. The changes in freestream velocity and effective angle of attack generate oscillatory airloads which excite the rigid body and elastic degrees of freedom of the rotor blades. These, in turn, couple (via the rotor hub and shaft) with the fuselage.

The reduction of these vibrations has always been a major task in helicopter design. Approaches can be made from the structural side, like dynamic tuning of the blade eigenmodes (aeroelastic tailoring), or from the aerodynamic side (e.g., utilization of less buffeting-sensitive blade sections). More recently methods to reduce blade vibrations by means of blade control inputs have been investigated: Application of higher harmonic control (HHC) or individual blade control (IBC) showed significant improvements [1-3].

Since the installations for rotor blade actuation are a part of the signal path from the controller to the rotor dynamic system, it is obvious that their dynamic characteristics must be taken into consideration for controller

design. Conventional pitch horn controlled rotors use hydraulic actuators to directly convert hydraulic fluid flow signals into blade pitch motions. Consequently at high frequencies, the pitch amplitude can be limited by the achievable fluid flow rates within the hydraulic installations, as is shown by Miao, et al. [1] and Walsh [2]. The findings of these surveys on HHC application on a S-76 helicopter indicated that, although pitch amplitudes of only 2 degrees were required to reach a certain reduction goal, only 1 degree could be realized. Unless one accepts the weight penalty of a more powerful hydraulic system, the effectiveness of HHC in conjunction with this rotor control system may be limited; a system more sensitive to control inputs could reduce or eliminate this limitation.

One way to increase sensitivity is to utilize the energy of the air flowing around the rotor blade for blade pitch changes by means of a small flap, located relatively far outboard at the trailing edge of the rotor blade. If the blade is not significantly restrained in its feathering motion by its link to the rotor hub, a deflection of the servo flap results in a change of aerodynamic moment about the blade feathering axis and therefore in a pitch change. Currently, this principle proves its practicability on the Kaman SH-2F helicopter with its servo flap controlled 101 Main Rotor (Figure 1). A servo flap of 6 lb is hinged at its quarter chord at 76% of the rotor radius, with its leading edge about half an inch behind trailing edge of the rotor blade (Figure 2). The flap is actuated by a system of control rods and cranks through the hollow rotor axis and the rotor blade (Figures 3 and 4). For future reference, it should be noted here that the positioning of the bob weight crank with respect to the flapping hinge and blade feathering axis provides for a mechanical

feedback of flapping and feathering into flap deflection (Figure 4; CMRB = Composite Main Rotor Blade).

If the tower rods are assumed to be fixed, the inboard arm of the bob weight crank is fixed in space. When the blade is deflected into positive pitch direction, θ, this point aft of the blade hinge line appears to move upward with respect to blade points at the same chordwise position, turning the crank in a clockwise direction. This corresponds to a positive perturbation in the servo flap deflection, δ (trailing edge down), or a positive feedback. On the other hand, since the crank is also located outboard of the flapping hinge, the same point seems to move downward with respect to the blade if the flapping angle, β, is increased, resulting in a decrease of the flap deflection angle. Positive flap deflection increases lift on the flap, and therefore adds negative aerodynamic moment, which reduces the blade pitch. Obviously, the negative flapping feedback is equivalent to a negative δ_3 - angle (for the actual 101 Rotor, -23.3°), which is counterbalanced by the restoring feathering feedback. The net effect of these two contradicting mechanical feedbacks depends on the size of the pitch-flap deflection feedback ratio, η, and the flapping-flap deflection feedback ratio, ζ.

The applicability of HHC on the 101 Rotor has been analytically investigated by Wei, et al. [3] using a combination of the Kaman 6F aeroelastic [4] and rotor multicyclic analysis (ROMULAN) [5] programs. Both programs were originally developed for another application of servo flap controls, the multicyclic controllable twist rotor, featuring both pitch horn and flap controls for 1/rev and higher harmonic inputs, respectively. Analytical findings of this rotor system were correlated with experimental investigations in the 40x80ft wind tunnel at NASA Ames [6].

6F is an aeroleastic performance code which accounts for the modes blade flapping, blade feathering, servo flap deflection, the first blade bending mode, and the first blade torsional mode. Flap and blade modes are coupled inertially and mechanically. Aerodynamics include reverse flow, stall, and Mach number effects. The dynamics of the rotor wake are modeled using a prescribed wake analysis. ROMULAN is a code based on the transfer matrix concept. That is, a linear relation is assumed between the system response variables, which in this case are stresses due to blade bending at several blade stations, or shears in transmission mount tubes, and control input variables. Here, these variables are the coefficients of the Fourier expansion of the servo flap input

$$\delta = \delta_0 + \delta_{1c} \cos \psi + \delta_{1s} \sin \psi + \delta_{2c} \cos 2\psi + \delta_{2s} \sin 2\psi + \ldots \qquad (1.1)$$

Then, a matrix of coefficients relates every control input parameter with each output parameter. These matrix elements are computed by ROMULAN using the simulation results of 6F, and control inputs can be calculated which eliminate higher harmonic stresses or 1/rev (peak-to-peak) stresses.

In case of the 101 Rotor, based on 1 degree HHC flap input amplitudes, hub shear and blade bending moment transfer functions have been determined with 6F. ROMULAN was then used to calculate 3/rev, 4/rev, and 5/rev HHC inputs in order to minimize the system response for three different forward flight speeds, so that a regression approach could be used to obtain control laws with velocity as a parameter. Finally, the effect of the calculated control inputs was simulated with 6F, showing significant reductions. As a comparison with the pitch-horn-controlled rotor, it is worth mentioning that the calculated input amplitudes were less than 1.25 degrees, and decreased for

higher input frequencies, underlining the higher sensitivity of the servo flap controlled 101 Rotor [3].

It appears that the servo flap control system offers promising opportunities for HHC applications. The procedure used by Kaman for HHC controller design [3] seems complicated and inflexible though. Improvements in this concern require a compact aeroelastic model of the combined system of blade aerodynamics, structural rotor dynamics and wake behavior which is flexible enough to provide a tool for different controller design methods.

1.2 Dynamic Wake Model

In general, aeroelastic analyses of a helicopter rotor include the three major components blade aerodynamics, rotor dynamics, and inflow feedback (Figure 5): given an angle of attack, an aerodynamic model yields the airloads on the rotor blade. A model of the rotor dynamics provides the response of the blade to these aerodynamic forces and moments. Blade motions, in turn, effect the aerodynamic environment of the blade and are therefore fed back into the aerodynamic model. Airloads, on the other hand, also influence the airflow directly, since the rotor blade sheds vorticity into the wake. This effect is a function of size and rate of change of airloads, so that the wake provides a frequency-dependent feedback. From an aeroelastic point of view, coupled blade aerodynamics and dynamics represent an open loop, quasi-steady model of the rotor (the so-called "inner problem"), not accounting for the wake effects, or, in other words, unsteady aerodynamics.

A praxis-oriented tool for aeroelastic analyses should provide enough flexibility to be applicable in either the frequency or the time domain. If this is accomplished, the same model can be used for controller design basing on Eigenvalue analysis (as compared to the elaborate method used by Kaman) and simulation of the response of the controller-rotor system to various control inputs and disturbances. A critical element under these premises is representation of wake dynamics. Theories modeling the wake response in form of a lift-deficiency function like the Theodorsen function for fixed-wing aircraft [7] can be eliminated immediately since they assume simple harmonic variations of the system states, are therefore only applicable to the stability boundary and do not provide information about the system damping apart from flutter initiation. Analyses of the system response to non-harmonic inputs in the time domain are not possible. Furthermore, applications of this approach to the complex wake geometry of a helicopter exclude forward flight, the most interesting condition for rotor dynamic analyses (like the work by Loewy [8]).

A more complex approach is to model the three-dimensional structure of trailing and tip vorticity shed by each rotor blade and to calculate resulting induced perturbation velocities on the rotor disc. Most classical rigid wake theories are not applicable to aeroelastic analyses since they assume an infinite number of blades. Free or distorted wake representations show a much higher degree of sophistication, which results in computational intensity. Most aeroelastic rotor programs (including Kaman's 6F) therefore include a prescribed wake geometry that has been determined by empirical or semi-empirical methods (flow visualization). Strictly speaking, prescribed wake results are only valid for the specific rotor configuration and flight condition

the wake geometry has been determined for and are of questionable applicability for other conditions.

Dynamic inflow formulations separate the blade lift problem from the reaction of the rotor wake as shown in Figure 1, providing greater flexibility in the choice of blade aerodynamic model and rotor configuration. The wake is represented as a dynamic system responding to changing pressures due to the passing rotor blades with a certain time delay as a result of inertia of the air. In the generalized dynamic wake theory of Peters and He [9][10], an acceleration potential is used to compute the induced inflow distribution over the rotor disc. This distribution is expressed by a finite number of state variables (or generalized coordinates), which resemble the coefficients of radial and azimuthal expansion functions. The wake model yields a system of first-order, ordinary differential equations and is therefore easily coupled with blade dynamics and can then be solved in the time or the frequency domain. Comparison of results obtained from coupling this wake represention with rigid blade dynamics of a four-bladed rotor in forward flight showed excellent correlation both with experimental inflow data and with predictions from both free and prescribed wake theories [10]. He [11] also used the model to calculate the time history of inflow and blade states. Su [12] finally showed its applicability to Eigenvalue analyses of rotors with one and four elastic blades in hover.

1.3 Scope of Work

Servo flap controlled rotors show promising characteristics for application of HHC or IBC systems due to their sensitivity. The existing analysis appears to be complex and inflexible, and is hence of limited benefit for extended parametric investigations and controller design. The development of a computational tool for these tasks is therefore desirable. This model should provide enough information to enable design and simulation of optimized controller-rotor geometry configurations (i.e., it should be applicable both in the time and the frequency domain). Furthermore, the code should require as little computer memory and computational effort (CPU time) as possible, with installation on personal computers as a goal.

The main emphasis in this work has therefore been, first, on the application of existing theories to rotors with a servo flap control system, and second on the development of a code that could serve as a research and design tool. As a first approach, the blades are assumed to be rigid in out-of-plane bending and torsion, and no inplane motion is considered. Blade and blade/flap quasi-steady aerodynamics and apparent mass terms are determined by thin airfoil theory, modified to account for flap contributions and correlated with two-dimensional test data. Due its simplicity and flexibility, the dynamic inflow model of Peters and He [9, 10] is used for modeling aerodynamic feedback via the rotor wake. It is applied for vertical flight conditions, but extension towards forward flight is sketched and the necessary expressions are provided. Proper function of the code are checked by program runs with Kaman 101 Rotor data, in which the response to five different HHC control input modes are calculated in hover.

CHAPTER 2

BLADE DYNAMICS

2.1 Determination of System Energies

In order to find the system of equations of motion in the three geometric state variables (flapping angle, β, feathering angle, θ, and servo flap deflection, δ), kinetic energy T and potential energy V of a rigid blade/flap rotor model have been determined. The first task in this process is to find kinetic energy

$$T = \int_{e}^{R} \int_{x_{le}}^{x_{te}} \mu \, |\{v\}|^2 \, dx \, dr \tag{2.1}$$

where e is the flapping hinge offset, R the rotor radius, x_{le} and x_{te} the chordwise coordinates of leading and trailing edge, respectively, μ the structural density and $|\{v\}|$ the absolute velocity. Fig. 6 shows the geometric arrangement of rotor blade and servo flap. The position vector $\{r\}$ to an arbitrary position (r,x) as a function of the three state variables β, θ, and δ is found to be

$$\{r\} = r\begin{Bmatrix}0\\1\\0\end{Bmatrix} + (r-e)\begin{Bmatrix}0\\\cos\beta\\\sin\beta\end{Bmatrix} + x\begin{Bmatrix}\cos\theta\\\sin\theta\sin\beta\\-\sin\theta\cos\beta\end{Bmatrix} + x^f\begin{Bmatrix}\cos(\theta+\delta)\\\sin(\theta+\delta)\sin\beta\\-\sin(\theta+\delta)\cos\beta\end{Bmatrix} \tag{2.2}$$

with the flapping hinge offset, e, radial coordinate, r, and the chordwise coordinates, x on the blade and x^f on the servo flap. The velocity vector of the observed particle is then

$$\{v\} = \frac{\partial\{r\}}{\partial t} + \begin{Bmatrix}0\\0\\\Omega\end{Bmatrix} \times \{r\} \tag{2.3}$$

with the rotor speed, Ω, so that

$$\begin{aligned}\{v\} = {} & \dot{\beta}\begin{Bmatrix}0\\-(r-e)\sin\beta + x\sin\theta\cos\beta + x^f\sin(\theta+\delta)\cos\beta\\(r-e)\cos\beta + x\sin\theta\sin\beta + x^f\sin(\theta+\delta)\sin\beta\end{Bmatrix}\\ & + \dot{\theta}\begin{Bmatrix}-x\sin\theta - x^f\sin(\theta+\delta)\\x\cos\theta\sin\beta + x^f\cos(\theta+\delta)\sin\beta\\-x\cos\theta\cos\beta - x^f\cos(\theta+\delta)\cos\beta\end{Bmatrix}\\ & + \dot{\delta}\begin{Bmatrix}-x^f\sin(\theta+\delta)\\x^f\cos(\theta+\delta)\sin\beta\\-x^f\cos(\theta+\delta)\cos\beta\end{Bmatrix}\\ & + \Omega\begin{Bmatrix}-e - (r-e)\cos\beta - x\sin\theta\sin\beta - x^f\sin(\theta+\delta)\sin\beta\\x\cos\theta - x^f\cos(\theta+\delta)\\0\end{Bmatrix}\end{aligned} \tag{2.4}$$

For the rotor blade, the position x^f and the terms multiplied by $\dot{\delta}$ and δ are without physical relevance, so that with equation (2.1) and the definitions (introducing the density of the blade, μ^b)

$$M^b = \int_e^R \int_{x_{le}}^{x_{te}} \mu^b \, dx \, dr$$

$$S_\beta^b = \int_e^R \int_{x_{le}}^{x_{te}} \mu^b \, (r-e) \, dx \, dr$$

$$S_\vartheta^b = \int_e^R \int_{x_{le}}^{x_{te}} \mu^b \, x \, dx \, dr$$

$$I_{\beta\beta}^b = \int_e^R \int_{x_{le}}^{x_{te}} \mu^b \, (r-e)^2 \, dx \, dr$$

$$I_{\beta\vartheta}^b = \int_e^R \int_{x_{le}}^{x_{te}} \mu^b \, (r-e) \, x \, dx \, dr$$

$$I_{\vartheta\vartheta}^b = \int_e^R \int_{x_{le}}^{x_{te}} \mu^b \, x^2 \, dx \, dr \qquad (2.5)$$

the kinetic energy of the rotor blade T^b becomes

$$
\begin{aligned}
T^b = {} & 1/2\, M^b e^2 \Omega^2 + S_\beta^b\, e \Omega^2 \cos\beta \\
& + 1/2\, I_{\beta\beta}^b \{ \dot\beta^2 + \Omega^2 \cos^2\beta \} \\
& + S_\theta^b\, e \{ \dot\theta\Omega \sin\theta + \Omega^2 \sin\theta \sin\beta \} \\
& + I_{\beta\theta}^b \{ \Omega^2 \sin\theta \sin\beta \cos\beta - \dot\beta (\dot\theta+\Omega) \cos\theta + \dot\theta\Omega \sin\theta \cos\beta \} \\
& + 1/2\, I_{\theta\theta}^b \{ \dot\theta^2 + \dot\beta^2 \sin^2\theta + \Omega^2(\sin^2\theta \sin^2\beta + \cos^2\theta) \\
& \qquad + 2\dot\beta\Omega \sin\theta \cos\theta \cos\beta + 2\dot\theta\Omega \sin\beta \}
\end{aligned} \tag{2.6}
$$

Equivalently, for the servo flap position x is equal to the chordwise distance from the blade hinge line to the flap hinge line, e_f , so that

$$
\begin{aligned}
T^f = {} & 1/2\, M^f \{ \dot\theta^2 e_f^2 + \dot\beta^2 e_f^2 \sin^2\theta + \Omega^2 [(e + e_f \sin\theta \sin\beta)^2 + e_f^2 \cos^2\theta] \\
& \qquad + 2\,\dot\theta\Omega [\, e_f^2 \sin^2\beta + e\, e_f \sin\theta\,] + 2\,\dot\beta\Omega\, e_f^2 \sin\theta \cos\theta \cos\beta \} \\
& + S_\beta^f \{ \Omega^2 (e + e_f \sin\theta \sin\beta) \cos\beta \\
& \qquad - \dot\beta\dot\theta\, e_f \cos\theta - \dot\beta\Omega\, e_f \cos\theta + \dot\theta\Omega\, e_f \sin\theta \cos\beta \} \\
& + 1/2\, I_{\beta\beta}^f \{ \dot\beta^2 + \Omega^2 \cos^2\beta \} \\
& + S_\delta^f \{ 1/4\, \dot\theta^2 e_f \sin 2\theta \sin(2\theta+\delta) + \dot\beta^2 e_f \sin\theta \sin(\theta+\delta) \\
& \qquad + \Omega^2 [\, e \sin(\theta+\delta) \sin\beta \\
& \qquad\qquad + e_f (\sin\theta \sin(\theta+\delta) \sin^2\beta + \cos\theta \cos(\theta+\delta)) \,] \\
& \qquad + (\dot\theta+\dot\delta)\, \Omega [\, e \sin(\theta+\delta) \sin\beta \\
& \qquad\qquad + e_f (\sin\theta \sin(\theta+\delta) + \cos\theta \cos(\theta+\delta)) \sin\beta \,] \\
& \qquad + \dot\beta\Omega\, e_f [(\sin(\theta+\delta) \cos\theta + \sin\theta \cos(\theta+\delta)) \cos\beta \,] \\
& \qquad + \dot\theta\dot\delta\, e_f [\, \sin(\theta+\delta) \sin\theta \cos(\theta+\delta) \cos\theta \,] \,\} \\
& + I_{\beta\delta}^f \{ \Omega^2 \sin(\theta+\delta) \sin\beta \cos\beta - \dot\beta (\dot\theta + \dot\delta + \Omega) \cos(\theta+\delta) \\
& \qquad + (\dot\theta+\dot\delta)\, \Omega \sin(\theta+\delta) \cos\beta \,] \,\} \\
& + 1/2\, I_{\delta\delta}^f \{ \dot\theta^2 + \dot\beta^2 \sin^2(\theta+\delta) + \dot\delta^2 + \Omega^2 (\sin^2(\theta+\delta) \sin^2\beta + \cos^2(\theta+\delta)) \\
& \qquad + 2\,(\dot\theta+\dot\delta)\Omega \sin\beta + 2\,\dot\beta\Omega \sin(\theta+\delta) \cos(\theta+\delta) \cos\beta + 2\dot\theta\dot\delta \}
\end{aligned} \tag{2.7}
$$

We take the mass density of the flap, μ^f, and the inboard and outboard limit positions of the blade/flap section r_2 and r_3 , respectively, (see Figure 2) in the following definitions:

$$M^f = \int_n \int_{x^f_{le}}^{x^f_{te}} \mu^f \, dx^f \, dr$$

$$S^f_\beta = \int_n \int_{x^f_{le}}^{x^f_{te}} \mu^f \, (r-e) \, dx^f \, dr$$

$$S^f_\delta = \int_n \int_{x^f_{le}}^{x^f_{te}} \mu^f \, x^f \, dx^f \, dr$$

$$I^f_{\beta\beta} = \int_n \int_{x^f_{le}}^{x^f_{te}} \mu^f \, (r-e)^2 \, dx^f \, dr$$

$$I^f_{\beta\delta} = \int_n \int_{x^f_{le}}^{x^f_{te}} \mu^f \, x^f \, (r-e) \, dx^f \, dr$$

$$I^f_{\delta\delta} = \int_n \int_{x^f_{le}}^{x^f_{te}} \mu^f \, x^{f2} \, dx^f \, dr \qquad (2.8)$$

Potential energy of this simplified model is easily found in deflections of the blade degrees of freedom from their reference positions $\bar{\beta}$ (precone) and $\bar{\theta}$ (pretwist of the blade retention straps for the 101 Rotor, see Figure 7) and in the concentrated hub springs in flapping, K_β, feathering, K_θ and a coupling term between both degrees of freedom, K_C:

$$V = 1/2\, K_\beta (\beta - \bar{\beta})^2 - K_c (\beta - \bar{\beta}) (\theta - \bar{\theta}) + 1/2\, K_\theta (\theta - \bar{\theta})^2 \qquad (2.9)$$

(The 101 Rotor features no flapping and coupling spring rates; these terms have been added for completeness and in view of future applications).

With equations (2.6), (2.7) and (2.9), the equations of motion in the three generalized coordinates β, δ, and θ take the well known form

$$\frac{d}{dt}\left(\frac{\partial(T^b + T^f)}{\partial \dot{\beta}}\right) - \frac{\partial(T^b + T^f)}{\partial \beta} - \frac{\partial V}{\partial \beta} = M_\beta^A$$

$$\frac{d}{dt}\left(\frac{\partial(T^b + T^f)}{\partial \dot{\theta}}\right) - \frac{\partial(T^b + T^f)}{\partial \theta} - \frac{\partial V}{\partial \theta} = M_\theta^A$$

$$\frac{d}{dt}\left(\frac{\partial(T^b + T^f)}{\partial \dot{\delta}}\right) - \frac{\partial(T^b + T^f)}{\partial \delta} - \frac{\partial V}{\partial \delta} = M_\delta^A \qquad (2.10)$$

The right hand side aerodynamic moments M_β^A, M_θ^A, and M_δ^A about flapping hinge, feathering axis and flap hinge line, respectively, remain to be specified in Chapter 3.

2.2 Linearization Considerations

Calculation of the derivatives in equation (2.10) yields rather lengthy expressions. It should be noted, that the derived system of equations is highly nonlinear and coupled, that is, it contains products of first time derivatives of generalized coordinates as well as trigonometric functions of the state variables themselves. The problem of linearization can be solved by application of a systematic ordering scheme. For the state variables themselves, it can be stated that - apart from instabilities - none of them will reach a value of 1 radian, so that

$$O[\alpha] = \varepsilon \tag{2.11}$$

where α can stand for each of the generalized coordinates (considering the absolute value of θ equation (2.11) does not hold since θ reaches 0.2 radian in hover trim; for the perturbation part however, sections 3.3 and 5.2 will show the correctness of assumption (2.11)). This allows small angle assumptions and also jusifies neglecting of quadratic terms in the states if all contributions $O[\varepsilon^2]$ and smaller are discarded:

$$\cos\alpha \approx 1, \quad \sin\alpha \approx \alpha, \quad \alpha\cdot\alpha \approx 0 \tag{2.12}$$

For simple harmonic excitation with multiples of the rotor speed $m\Omega$, the amplitude of the first time derivative of each state variable $\tilde{\dot{\alpha}}$ has a distinct relation to the amplitude of the state itself:

$$\frac{\tilde{\dot{\alpha}}}{\Omega} = m\,\tilde{\alpha} \tag{2.13}$$

The largest amplitude $\tilde{\alpha}$ occurs around m = 1 and is O[ε]; for the pure mechanical system, it then also seems reasonable to assume that $O[m\,\tilde{\alpha}] = \varepsilon$, so that

$$O\left[\frac{\tilde{\dot{\alpha}}}{\Omega}\right] = \varepsilon \tag{2.14}$$

Discarding of terms smaller or equal $O[\varepsilon^2]$ leads then to elimination of expressions multiplied by $\dot{\alpha}\,\dot{\alpha}$.

A purely mechanical system without dampers cannot have velocity-proportional (i.e., damping-like) terms; it seems therefore reasonable to linearize the equations of motion (2.10) by neglect of expressions factored by $\dot{\alpha}\,\Omega$, although $O[\dot{\alpha}\,\Omega] = \varepsilon$.

2.3 Equations of Motion

Under the two assumptions of section 2.2, equation (2.10) takes the following form:

Flapping:

$$[\, I^b_{\beta\beta} + I^f_{\beta\beta} \,]\, \ddot{\beta} + \Omega^2 [\, I^b_{\beta\beta} + I^f_{\beta\beta} + (S^b_\beta + S^f_\beta)\, e + K_\beta / \Omega^2 \,]\, \beta$$

$$- [\, I^b_{\beta\theta} + I^f_{\beta\delta} + S^f_\beta e_f \,]\, \ddot{\theta} - \Omega^2 [\, I^b_{\beta\theta} + I^f_{\beta\delta} + (S^b_\theta + S^f_\delta)\, e + S^f_\beta e_f + M^f e\, e_f + K_c / \Omega^2 \,]\, \theta$$

$$- I^f_{\beta\delta}\, \ddot{\delta} - \Omega^2 [\, I^f_{\beta\delta} + S^f_\delta e \,]\, \delta$$

$$= M^A_\beta + K_\beta \bar{\beta} - K_c \bar{\theta} \qquad (2.15)$$

Feathering:

$$- [\, I^b_{\beta\theta} + I^f_{\beta\delta} + S^f_\beta e_f \,]\, \ddot{\beta} - \Omega^2 [\, I^b_{\beta\theta} + I^f_{\beta\delta} + (S^b_\theta + S^f_\delta)\, e + S^f_\beta e_f + M^f e\, e_f + K_c / \Omega^2 \,]\, \beta$$

$$+ [\, I^b_{\theta\theta} + I^f_{\delta\delta} + M^f e_f^2 \,]\, \ddot{\theta} + \Omega^2 [\, I^b_{\theta\theta} + I^f_{\delta\delta} + 2\, S^f_\delta e_f + M^f e_f^2 + K_\theta / \Omega^2 \,]\, \theta$$

$$+ I^f_{\delta\delta}\, \ddot{\delta} + \Omega^2 [\, I^f_{\delta\delta} + S^f_\delta e_f \,]\, \delta$$

$$= M^A_\theta + K_\theta \bar{\theta} - K_c \bar{\beta} \qquad (2.16)$$

Servo flap deflection:

$$- I^f_{\beta\delta}\,\ddot{\beta} - \Omega^2\,[\,I^f_{\beta\delta} + S^f_\delta\, e\,]\;\beta$$

$$+ I^f_{\delta\delta}\;\ddot{\theta} + \Omega^2\,[\,I^f_{\delta\delta} + S^f_\delta\, e_f\,]\;\theta$$

$$+ I^f_{\delta\delta}\,\ddot{\delta} + \Omega^2\,[\,I^f_{\delta\delta} + S^f_\delta\, e_f\,]\;\delta$$

$$= M^A_\delta \tag{2.17}$$

Introducing the nondimensional time $\bar{t}$ (equivalent to the azimuthal position ψ of the blade)

$$\bar{t} = \psi = \Omega t \tag{2.18}$$

and dividing the whole formula by $I_{\beta\beta}\,\Omega^2$ ($I_{\beta\beta} = I_{\beta\beta}{}^b + I_{\beta\beta}{}^f$) yields the final form of the equations of motion in their physical coordinates in matrix notation:

$$\begin{bmatrix} 1 & -\bar{I}_{\beta\theta} & -\bar{I}_{\beta\delta} \\ -\bar{I}_{\beta\theta} & \bar{I}_{\theta\theta} & \bar{I}_{\theta\delta} \\ -\bar{I}_{\beta\delta} & \bar{I}_{\theta\delta} & \bar{I}_{\delta\delta} \end{bmatrix} \begin{Bmatrix} \beta \\ \theta \\ \delta \end{Bmatrix}'' + \begin{bmatrix} \bar{K}_{\beta\beta} & -\bar{K}_{\beta\theta} & -\bar{K}_{\beta\delta} \\ -\bar{K}_{\beta\theta} & \bar{K}_{\theta\theta} & \bar{K}_{\theta\delta} \\ -\bar{K}_{\beta\delta} & \bar{K}_{\theta\delta} & \bar{K}_{\delta\delta} \end{bmatrix} \begin{Bmatrix} \beta \\ \theta \\ \delta \end{Bmatrix}$$

$$= \begin{Bmatrix} \gamma \int_{\bar{r}_0}^{1} \bar{L}/(a\bar{c})\;(\bar{r}-\bar{e})\,d\bar{r} + \bar{K}_\beta\,\bar{\beta} - \bar{K}_c\,\bar{\theta} \\ \gamma \int_{\bar{r}_0}^{1} \bar{M}/(a\bar{c})\,d\bar{r} \qquad\qquad - \bar{K}_c\,\bar{\beta} + \bar{K}_\theta\,\bar{\theta} \\ \bar{M}_\delta \end{Bmatrix} \tag{2.19}$$

The right hand side of the equations contains aerodynamic and mechanical forces acting on the system, including the integrals of blade and blade/flap nondimensional sectional lift $\bar{L}$ and moment $\bar{M}$ as a function of the two-dimensional lift curve slope a, $\bar{K}_\beta$, $\bar{K}_c$ and $\bar{K}_\theta$ are nondimensional spring rates, $\bar{c}$, $\bar{e}$, $\bar{r}$, and $\bar{r}_0$ are the blade mean chord, flapping hinge offset, radial position and root cutout, respectively, all normalized by the rotor radius R. The elements of the mass matrix are

$$\bar{I}_{\beta\theta} = (I^b_{\beta\theta} + I^f_{\beta\delta} + S^f_\beta e_f) / I_{\beta\beta}$$

$$\bar{I}_{\beta\delta} = I^f_{\beta\delta} / I_{\beta\beta}$$

$$\bar{I}_{\delta\delta} = \bar{I}_{e\delta} = I^f_{\delta\delta} / I_{\beta\beta}$$

$$\bar{I}_{e\theta} = (I^b_{e\theta} + I^f_{\delta\delta} + M^f e_f^2) / I_{\beta\beta} \qquad (2.20)$$

The stiffness matrix has the components

$$\bar{K}_{\beta\beta} = 1 + [(S^b_\beta + S^f_\beta) e + K_\beta / \Omega^2] / I_{\beta\beta}$$

$$\bar{K}_{\beta\theta} = \bar{I}_{\beta\theta} + [(S^b_\theta + S^f_\delta + M_f e_f) e + K_c / \Omega^2] / I_{\beta\beta}$$

$$\bar{K}_{\beta\delta} = \bar{I}_{\beta\delta} + S^f_\delta e / I_{\beta\beta}$$

$$\bar{K}_{e\theta} = \bar{I}_{e\theta} + [2 S^f_\delta e_f + K_\theta / \Omega^2] / I_{\beta\beta}$$

$$\bar{K}_{\delta\delta} = \bar{I}_{\delta\delta} + S^f_\delta e_f / I_{\beta\beta} \qquad (2.21)$$

2.3 Choice of Dynamic Model

All derivations thus far have been made without consideration of any control system kinematics or dynamics in order to keep the model as general as possible. The system considered up to now, with direct servo flap input, will be denoted as system I. It leaves open the possibility of investigation of flap actuation mechanisms other than the one described in section 1.1; for higher harmonic control electrical actuators mounted in the rotor blades could be considered, for example. In this case, the control input is $\delta_c = \delta$.

The control linkages on the 101 Rotor, however, are a dynamic system on their own. Neglecting damping effects, the moment on the servo flap $\overline{M}_\delta$ in equation (2.19) can be expressed as a function of normalized equivalent control stiffness $\overline{K}_{control}$ and inertia $\overline{I}_{control}$

$$\overline{M}_\delta = -\overline{I}_{control}\,\delta^{**} - \overline{K}_{control}\,(\delta - \hat{\delta}) \tag{2.22}$$

where

$$\hat{\delta} = \delta_c + \eta\beta + \zeta\theta \tag{2.23}$$

with the two feedback ratios η and ζ. Replacing $\overline{M}_\delta$ in (2.19) yields the equations of motion in β, θ, and δ as a function of the control input δ_c (system II) , noting that $\overline{L}$ and $\overline{M}$ are functions of the blade states and therefore provide aerodynamical feedback.

$$\begin{bmatrix} 1 & -\bar{I}_{\beta\theta} & -\bar{I}_{\beta\delta} \\ -\bar{I}_{\beta\theta} & \bar{I}_{\theta\theta} & \bar{I}_{\theta\delta} \\ -\bar{I}_{\beta\delta} & \bar{I}_{\theta\delta} & \bar{I}_{\delta\delta} + \bar{I}_{control} \end{bmatrix} \begin{Bmatrix} \beta \\ \theta \\ \delta \end{Bmatrix}^{**}$$

$$+ \begin{bmatrix} \bar{K}_{\beta\beta} & -\bar{K}_{\beta\theta} & -\bar{K}_{\beta\delta} \\ -\bar{K}_{\beta\theta} & \bar{K}_{\theta\theta} & \bar{K}_{\theta\delta} \\ -\bar{K}_{\beta\delta} - \eta\,\bar{K}_{control} & \bar{K}_{\theta\delta} - \zeta\,\bar{K}_{control} & \bar{K}_{\delta\delta} + \bar{K}_{control} \end{bmatrix} \begin{Bmatrix} \beta \\ \theta \\ \delta \end{Bmatrix}$$

$$= \begin{Bmatrix} \gamma \int_{\bar{r}_0}^{1} \bar{L}/(a\bar{c})\ (\bar{r} - \bar{e})\, d\bar{r} + \bar{K}_\beta\,\bar{\beta} - \bar{K}_c\,\bar{\theta} \\ \gamma \int_{\bar{r}_0}^{1} \bar{M}/(a\bar{c})\, d\bar{r} \qquad - \bar{K}_c\,\bar{\beta} + \bar{K}_\theta\,\bar{\theta} \\ \bar{K}_{control}\,\delta_c \end{Bmatrix} \tag{2.24}$$

A third possibility is to account for control system kinematics neglecting its dynamics, i.e. $\bar{K}_{control} \Rightarrow \infty$ and $\bar{I}_{control} \Rightarrow 0$, so that $\hat{\delta} = \delta_c$, and system III becomes

$$\begin{bmatrix} 1 - \eta\,\bar{I}_{\beta\delta} & -\bar{I}_{\beta\theta} - \zeta\,\bar{I}_{\beta\delta} & -\bar{I}_{\beta\delta} \\ -\bar{I}_{\beta\theta} + \eta\,\bar{I}_{\theta\delta} & \bar{I}_{\theta\theta} + \zeta\,\bar{I}_{\theta\delta} & \bar{I}_{\theta\delta} \\ -\bar{I}_{\beta\delta} + \eta\,\bar{I}_{\delta\delta} & \bar{I}_{\theta\delta} + \zeta\,\bar{I}_{\delta\delta} & \bar{I}_{\delta\delta} \end{bmatrix} \begin{Bmatrix} \beta \\ \theta \\ \delta_c \end{Bmatrix}^{**}$$

$$+ \begin{bmatrix} \bar{K}_{\beta\beta} - \eta\,\bar{K}_{\beta\delta} & -\bar{K}_{\beta\theta} - \zeta\,\bar{K}_{\beta\delta} & -\bar{K}_{\beta\delta} \\ -\bar{K}_{\beta\theta} + \eta\,\bar{K}_{\theta\delta} & \bar{K}_{\theta\theta} + \zeta\,\bar{K}_{\theta\delta} & \bar{K}_{\theta\delta} \\ -\bar{K}_{\beta\delta} + \eta\,\bar{K}_{\delta\delta} & \bar{K}_{\theta\delta} + \zeta\,\bar{K}_{\delta\delta} & \bar{K}_{\delta\delta} \end{bmatrix} \begin{Bmatrix} \beta \\ \theta \\ \delta_c \end{Bmatrix}$$

$$= \begin{Bmatrix} \gamma \int_{\bar{r}_0}^{1} \bar{L}/(a\bar{c})\ (\bar{r} - \bar{e})\, d\bar{r} + \bar{K}_\beta\,\bar{\beta} - \bar{K}_c\,\bar{\theta} \\ \gamma \int_{\bar{r}_0}^{1} \bar{M}/(a\bar{c})\, d\bar{r} \qquad - \bar{K}_c\,\bar{\beta} + \bar{K}_\theta\,\bar{\theta} \\ \bar{M}_\delta^{A} \end{Bmatrix} \tag{2.25}$$

For all three open-loop systems (no aerodynamics) the eigenvalues were determined to compare these simple models with the more complex systems used by Wei and Jones [13] (which include first blade bending and twisting mode). Rotor data for this and all other applications in this work have been taken from Fitzpatrick, et al. [14], a summary of which can be found in Appendix A. The results, listed in Table 1, show acceptable correlation (for system I and III the servo flap cannot be considered as a degree of freedom, but rather a prescribed control input, which is to be fixed for this analysis, so that no eigenvalue can be computed). System II comes closest to the analyses of [13], since δ is considered as a degree of freedom. The control system stiffness $\bar{K}_{control}$ and inertia $\bar{I}_{control}$ were not available; they were matched to yield the uncoupled eigenvalue of the servo flap degree of freedom (14.72/rev, 6F analysis [13]). It was found that the additional control system inertia could be neglected. Stiffness for this case was found to be $\bar{K}_{control} = 0.001362$, a value that is of the same order as e.g. the feathering stiffness $\bar{K}_{\theta\theta}$. Since moments on the servo flap will be considerably smaller than those on the blade, a relatively small effect of the control rod elasticity is to be expected. This suggests the use of system III and justifies its assumptions. In fact, a comparison of the modal matrices

$$[X]_I = \begin{bmatrix} 1.0 & 0.00214896 \\ 0.314741 & 1.0 \end{bmatrix}$$

$$[X]_{II} = \begin{bmatrix} 1.0 & 0.00214291 & 0.00013896 \\ 0.2150770 & 1.0 & 0.00843544 \\ 0.0976602 & 0.00550106 & 1.0 \end{bmatrix}$$

$$[X]_{III} = \begin{bmatrix} 1.0 & 0.00214433 \\ 0.213121 & 1.0 \end{bmatrix} \qquad (2.26)$$

leads to the conclusion that the dynamics of the control system have negligible influence on the blade/flap eigenmodes (2.26). It was therefore decided to use system III for the analysis of the coupled blade dynamics-aerodynamics-inflow dynamics model in Chapter 5.

CHAPTER 3

ROTOR BLADE AND BLADE/FLAP AERODYNAMICS

3.1 General Considerations

From an aerodynamic point of view, the main difference between a servo flap controlled rotor and a conventional rotor is the application of a flapped - or multi-component - section in the region of highest airload generation. In case of the 101 Rotor, a symmetric flap is located aft of a modified NACA 23012 section, with a slot of about 2% of the blade chord. Two-dimensional data from steady wind tunnel tests on three quarter scale models [14] show that this arrangement leads to nonlinear lift and moment characteristics in regions of unseparated flow. Figures 8 and 9 are plots of test data for Mach numbers around 0.5, corresponding to the conditions at the servo flap (76%R) in hover at sea level, standard day (M = 0.48). The most obvious characteristic of the c_l plot are two positive steps for negative lift coefficients. With decreasing servo flap deflection (δ is negative in normal operation), the stepsize increases and the steps initiate at larger angles of attack. Since the unflapped section still shows linear characteristics in this region, it is obvious that the nonlinearities are a result of aerodynamic

interaction between the two components. Most probably passing of the flap through the blade's boundary layer is the reason for this behavior, supported by the observation that these nonlinearities depend both on the angle of attack of the configuration and on the flap deflection angle. A more detailed analysis of this phenomenon could be performed (e.g., by the methodology of Stevens, et al. [15]; being a two-dimensional model though, it lacks modeling of three-dimensional effects like radial flow influence, which effect boundary layer thickness and lift generation on rotating airfoils [16]). Investigations of this kind, however, are beyond the scope of this work, and a complex CFD model does not lend itself to compact aeroelastic analyses.

In order to obtain a representation of the rotor blade and blade/flap aerodynamics which matches the requirements for the intended application, simplifications have to be made. It is useful to note that the nonlinear effects occur in negative lift conditions and decrease with increasing flap deflection. Quasi-steady operation can therefore still be assumed to occur in a linear region. This assumption simplifies the complete model significantly.

In this work, a general approach by Johnson [17] has been used to obtain lift and moment expressions for both the blade and the flap They include quasi-steady and apparent mass terms due to blade motion, and contain inflow explicitly, as required for application of He and Peters' wake theory. Quasi-steady aerodynamics of the blade/flap configuration have then been modeled using coefficients for lift, moment and servo flap influence obtained from a regression analysis of test data (linear region). Apparent mass terms for blade and flap separately are then combined to yield the unsteady partitions of the lift and moment expressions.

3.2 Lift and Moment Expressions

According to Johnson [17], downwash at a rotor blade section can be written as a function of a mean value A and a term linear in the chordwise coordinate $\bar{x}$, multiplied by the effective pitch rate B of the blade, and the rotor inflow λ:

$$\bar{w} = A - \lambda + B\,\bar{x} \tag{3.1}$$

A is the effective angle of attack of a blade section for a certain azimuthal position ψ of the blade, multiplied by freestream velocity, or downwash at the sectional axis of rotation (i.e. blade feathering axis)

$$A = -\vec{z_0} + (\bar{r} + \mu \sin\psi)\,\theta - \mu \cos\psi \frac{d\bar{z}_0}{d\bar{r}} \tag{3.2}$$

and consists of the out-of-plane velocity of the blade feathering axis, freestream velocity components normal to the blade surface (as a function of radial coordinate $\bar{r}$ and advance ratio μ) and inplane radial flow velocity, respectively. The effective pitch rate contains contributions from blade pitch rate, blade slope, and twist:

$$B = \theta^{*} + \frac{d\bar{z}_0}{d\bar{r}} + \mu \cos\psi \frac{d\theta}{d\bar{r}} \tag{3.3}$$

Neglecting terms higher than O[$\bar{c}$] in lift and higher than O[$\bar{c}^2$] in moment, sectional loads are finally found to be

$$\frac{\bar{L}}{a\bar{c}} = \frac{1}{2}(\bar{r} + \mu \sin\psi)\,\bar{w}|_{3/4\,\bar{c}} + \frac{\bar{c}}{8}\left[\bar{w}^{*} + \mu\cos\psi\frac{d\bar{w}}{d\bar{r}}\right] \tag{3.4}$$

$$\frac{\bar{M}}{a\bar{c}} = -\bar{c}\,\bar{e}_A\frac{\bar{L}}{a\bar{c}} - \frac{\bar{c}^2}{32}\left[(\bar{r} + \mu\sin\psi)B + \bar{w}^{*} + \mu\cos\psi\frac{d\bar{w}}{d\bar{r}}\right] \tag{3.5}$$

where $\bar{e}_A$ is the chordwise distance from the feathering axis or hinge line to the aerodynamic center. For unflapped sections of a rigid rotor blade the vertical coordinate of the blade hinge line is

$$\bar{z}_0 = \beta(\bar{r} - \bar{e}) \tag{3.6}$$

Introducing the sectional lift coefficient at zero pitch angle, $c_{l0,b}$, and separating quasi-steady, non-circulatory, and radial flow contributions, the expanded lift expressions take the form

$$\begin{aligned}
\frac{\vec{L}_{qs}^{b}}{a\bar{c}} &= -\frac{1}{2}(\bar{r} - \bar{e})(\bar{r} + \mu\sin\psi)\beta^{*} + \frac{3}{8}\bar{c}(\bar{r} + \mu\sin\psi)\theta^{*} \\
&\quad + \left(\frac{3}{8}\bar{c} + \mu\cos\psi\right)(\bar{r} + \mu\sin\psi)\beta + \frac{1}{2}(\bar{r} + \mu\sin\psi)^2\theta \\
&\quad + \frac{1}{2}(\bar{r} + \mu\sin\psi)^2\frac{c_{l0,b}}{a} + \frac{3}{8}\bar{c}\,\mu\cos\psi(\bar{r} + \mu\sin\psi)\frac{d\theta}{d\bar{r}} \\
&\quad - \frac{1}{2}(\bar{r} + \mu\sin\psi)\lambda \\
\frac{\vec{L}_{nc}^{b}}{a\bar{c}} &= \frac{\bar{c}}{8}\left[-(\bar{r} - \bar{e})\beta^{**} - \mu\cos\psi\,\beta^{*} + (\bar{r} + \mu\sin\psi)\theta^{*} + \mu\sin\psi\,\beta + \mu\cos\psi\,\theta\right] \\
\frac{\vec{L}_{rad}^{b}}{a\bar{c}} &= \frac{\bar{c}}{8}\left[-\mu\cos\psi(\beta^{*} - \theta) + \mu\cos\psi(\bar{r} + \mu\sin\psi)\frac{d\theta}{d\bar{r}}\right]
\end{aligned} \tag{3.7}$$

Similarly, the sectional aerodynamic moment components are

$$\frac{\vec{M}^{b}_{qs}}{a\,\bar{c}} = -\frac{\bar{c}\bar{e}_A}{2}(\bar{r}-\bar{e})(\bar{r}+\mu\sin\psi)\beta^{*} - \frac{\bar{c}^2}{8}(\bar{e}_A+\frac{1}{4})(\bar{r}+\mu\sin\psi)\theta^{*}$$

$$+\left[\frac{\bar{c}^2}{8}(\bar{e}_A+\frac{1}{4})+\frac{\bar{c}\,\bar{e}_A}{2}\mu\cos\psi\right](\bar{r}+\mu\sin\psi)\beta - \frac{\bar{c}\,\bar{e}_A}{2}(\bar{r}+\mu\sin\psi)^2\theta$$

$$+\frac{\bar{c}}{2}(\bar{r}+\mu\sin\psi)^2\frac{c^{b}_{m\alpha}}{a} - \frac{\bar{c}^2}{8}(3\bar{e}_A+\frac{1}{4})\mu\cos\psi\frac{d\theta}{d\bar{r}} + \frac{\bar{c}\,\bar{e}_A}{2}(\bar{r}+\mu\sin\psi)\lambda$$

$$\frac{\vec{M}^{b}_{nc}}{a\,\bar{c}} = \frac{\bar{c}^2}{32}(\frac{1}{4}+\bar{e}_A)\left[-(\bar{r}-\bar{e})\beta^{**} + \mu\cos\psi\,\beta^{*} - (\bar{r}+\mu\sin\psi)\theta^{*}\right.$$

$$\left.-\mu\sin\psi\,\beta - \mu\cos\psi\,\theta\right]$$

$$\frac{\vec{M}^{b}_{rad}}{a\,\bar{c}} = \frac{\bar{c}^2}{32}(\frac{1}{4}+\bar{e}_A)\left[\mu\cos\psi\,\beta^{*} - \mu\cos\psi\,\theta - \mu\cos\psi(\bar{r}+\mu\sin\psi)\frac{d\theta}{d\bar{r}}\right] \qquad (3.8)$$

The same approach can be followed for the flap, where

$$\bar{z}_0 = \beta\bar{r} - \bar{c}\,\bar{e}_f\theta \qquad (3.9)$$

and moment is to be calculated with respect to the blade feathering axis. Therefore, $\bar{e}_A$ in (3.5) must be replaced by the distance $\bar{e}_f$ between the flap hinge line and the blade feathering axis. Combining these terms with expressions for the flapped rotor blade part yields apparent mass and radial flow contribution of the total sectional loads.

In order to correlate the aerodynamic model with test data, quasi-steady lift and moment terms have not been calculated in this matter. To account for additional airloads produced by the flap in the flapped section of the rotor blade, lift and moment coefficient derivatives with respect to servo flap deflection were taken from a regression analysis of data in Reference [14], yielding linear relations for blade and blade/flap lift and moment coefficients:

$$
\begin{aligned}
c_m^b &= c_{m0}^b + c_{m\alpha}^b \, \alpha \\
c_m^{bf} &= c_{m0}^{bf} + c_{m\alpha}^{bf} \, \alpha + c_{m\delta} \, \delta \\
c_l^b &= c_{l0}^b + a \, \alpha \\
c_l^{bf} &= c_{l0}^{bf} + \kappa^{bf} \, a \, \alpha + c_{l\delta} \, \delta
\end{aligned}
\tag{3.10}
$$

where α is the effective angle of attack, and κ^{bf} is the ratio of the lift curve slopes of flapped to unflapped rotor section. Numerical values are listed in Appendix A, and correlation of this linear regression with test data is shown in Figures 10 and 11. Values for $c_{m\alpha}$ have been calculated but not used later on, since the dependency of the moment on angle of attack is accounted for by the lift term in the moment equation (3.5). The lengthy results of the described procedure are shown below.

Sectional lift:

$$\frac{\bar{L}_{qs}^{bf}}{a\bar{c}} = -\frac{1}{2}\kappa^{bf}(\bar{r}-\bar{e})(\bar{r}+\mu\sin\psi)\beta^{*} + \frac{3}{8}\kappa^{bf}\bar{c}(\bar{r}+\mu\sin\psi)\theta^{*}$$

$$+\kappa^{bf}(\frac{3}{8}\bar{c}+\frac{1}{2}\mu\cos\psi)(\bar{r}+\mu\sin\psi)\beta + \frac{1}{2}\kappa^{bf}(\bar{r}+\mu\sin\psi)^{2}\theta$$

$$+\frac{1}{2}(\bar{r}+\mu\sin\psi)^{2}\frac{c_{l\delta}}{a} + \frac{3}{8}\kappa^{bf}\bar{c}\mu\cos\psi(\bar{r}+\mu\sin\psi)\frac{d\theta}{d\bar{r}}$$

$$-\frac{1}{2}\kappa^{bf}(\bar{r}+\mu\sin\psi)\lambda$$

$$\frac{\bar{L}_{nc}^{bf}}{a\bar{c}} = (\frac{\bar{c}}{8}\kappa^{bf}+\frac{\bar{c}_f}{8}\kappa^{f})[-(\bar{r}-\bar{e})\beta^{**} - \mu\cos\psi\beta^{*}$$

$$+(\bar{r}+\mu\sin\psi)\theta^{*}+\mu\sin\psi\beta]$$

$$+\frac{\bar{c}_f}{8}\kappa^{f}[\bar{c}\,\bar{e}_f\theta^{**}+\mu\cos\psi\theta$$

$$+(\bar{r}+\mu\sin\psi)\delta^{*}+\mu\cos\psi\delta]$$

$$\frac{\bar{L}_{rad}^{bf}}{a\bar{c}} = (\frac{\bar{c}}{8}\kappa^{bf}+\frac{\bar{c}_f}{8}\kappa^{f})[-\mu\cos\psi(\beta^{*}-\theta)+\mu\cos\psi(\bar{r}+\mu\sin\psi)\frac{d\theta}{d\bar{r}}]$$

$$+\frac{\bar{c}_f}{8}\kappa^{f}[\mu\cos\psi\delta] \qquad (3.11)$$

Sectional moment:

$$\frac{\overline{M}^{bf}_{qs}}{a\,\bar{c}} = -\frac{\bar{c}\,\bar{e}_A}{2}\,\kappa^{bf}(\bar{r}-\bar{e})(\bar{r}+\mu\sin\psi)\,\beta^{*}$$

$$-\frac{\bar{c}^2}{8}\,\kappa^{bf}\left(\bar{e}_A+\frac{1}{4}\right)(\bar{r}+\mu\sin\psi)\,\theta^{*}$$

$$+\left[\frac{\bar{c}^2}{8}\left(\bar{e}_A+\frac{1}{4}\right)+\frac{\bar{c}\,\bar{e}_A}{2}\mu\cos\psi\right]\kappa^{bf}(\bar{r}+\mu\sin\psi)\,\beta$$

$$-\frac{\bar{c}\,\bar{e}_A}{2}\,\kappa^{bf}(\bar{r}+\mu\sin\psi)^2\,\theta+\frac{\bar{c}}{2}(\bar{r}+\mu\sin\psi)^2\,\frac{c^{bf}_{m\,0}}{a}$$

$$-\frac{\bar{c}^2}{8}\,\kappa^{bf}\left(3\,\bar{e}_A+\frac{1}{4}\right)\mu\cos\psi\,\frac{d\theta}{d\bar{r}}+\frac{\bar{c}}{2}(\bar{r}+\mu\sin\psi)^2\,\frac{c_{m\delta}}{a}\,\delta$$

$$+\frac{\bar{c}\,\bar{e}_A}{2}\,\kappa^{bf}(\bar{r}+\mu\sin\psi)\,\lambda$$

$$\frac{\overline{M}^{bf}_{nc}}{a\,\bar{c}} = \left[\frac{\bar{c}^2}{32}\,\kappa^{bf}\left(\frac{1}{4}+\bar{e}_A\right)+\frac{\bar{c}_f^2}{32}\,\kappa^{f}\left(\frac{1}{4}+\frac{\bar{e}_f}{\zeta_f}\right)\right]$$

$$\left[-(\bar{r}-\bar{e})\,\beta^{**}+\mu\cos\psi\,\beta^{*}-(\bar{r}+\mu\sin\psi)\,\theta^{*}-\mu\sin\psi\,\beta-\mu\cos\psi\,\theta\right]$$

$$-\frac{\bar{c}_f^2}{32}\,\kappa^{f}\left(\frac{1}{4}+\frac{\bar{e}_f}{\zeta_f}\right)\left[\theta^{**}+(\bar{r}+\mu\sin\psi)\,\delta^{*}+\mu\cos\psi\,\delta\right]$$

$$\frac{\overline{M}^{bf}_{rad}}{a\,\bar{c}} = \left[\frac{\bar{c}^2}{32}\,\kappa^{bf}\left(\frac{1}{4}+\bar{e}_A\right)+\frac{\bar{c}_f^2}{32}\,\kappa^{f}\left(\frac{1}{4}+\frac{\bar{e}_f}{\zeta_f}\right)\right]$$

$$\left[\mu\cos\psi\,\beta^{*}-\mu\cos\psi\,\theta-\mu\cos\psi\,(\bar{r}+\mu\sin\psi)\,\frac{d\theta}{d\bar{r}}\right]$$

$$-\frac{\bar{c}_f^2}{32}\,\kappa^{f}\left(\frac{1}{4}+\frac{\bar{e}_f}{\zeta_f}\right)\left[\mu\cos\psi\,\delta\right] \qquad (3.12)$$

3.3 Coupling with Rotor Dynamics

In order to investigate response of the rotor blade without consideration of inflow dynamics, the right hand sides of all all of the three systems defined in Chapter 2 were replaced by integrals of lift and moment expressions (as derived in the preceding section) over the rotor radius. These integrals consist of four intervals:

$r_0 < r < r_1$: blade aerodynamics, twist rate $d\theta/d\bar{r} = \theta_{T1}$

$r_1 < r < r_2$: blade aerodynamics, twist rate $d\theta/d\bar{r} = \theta_{T2}$

$r_2 < r < r_3$: blade/flap aerodynamics, twist rate $d\theta/d\bar{r} = \theta_{T2}$

$r_3 < r < R$: blade aerodynamics, twist rate $d\theta/d\bar{r} = \theta_{T2}$

Since the flight condition was hover, all terms multiplied by the advance ratio μ and all radial flow components disappear. Since they will be used in Chapter 5, the remaining expressions for aerodynamic moments about the flapping hinge $\overline{M}_\beta^A$ and about the feathering axis $\overline{M}_\theta^A$ (from Equation (2.10)) are listed below.

Flapping:

$$\overline{M_\beta^A} = \gamma \int_{\bar{n}}^{1} \frac{\bar{L}}{a\,\bar{c}} (\bar{r} - \bar{e})\, d\bar{r}$$

$$= \gamma \{ [\frac{\bar{c}}{8} (\int_{\bar{n}}^{1} (\bar{r} - \bar{e})^2 d\bar{r} + (\kappa^{bf} - 1) \int_{\bar{n}}^{\bar{n}} (\bar{r} - \bar{e})^2 d\bar{r})$$

$$\frac{\bar{c}_f}{8} \kappa^f \int_{\bar{n}}^{\bar{n}} (\bar{r} - \bar{e})^2 d\bar{r}] \beta^{**}$$

$$+ [\frac{\bar{c}_f}{8} \kappa^f \bar{c}\, \bar{e}_f \int_{\bar{n}}^{\bar{n}} (\bar{r} - \bar{e}) d\bar{r}] \theta^{**}$$

$$\frac{1}{2} [\int_{\bar{n}}^{1} \bar{r} (\bar{r} - \bar{e})^2 d\bar{r} + (\kappa^{bf} - 1) \int_{\bar{n}}^{\bar{n}} \bar{r} (\bar{r} - \bar{e})^2 d\bar{r}] \beta^{*}$$

$$+ \frac{1}{8} [\bar{c} (3 + 1) (\int_{\bar{n}}^{1} \bar{r} (\bar{r} - \bar{e}) d\bar{r} + (\kappa^{bf} - 1) \int_{\bar{n}}^{\bar{n}} \bar{r} (\bar{r} - \bar{e}) d\bar{r}]$$

$$+ \bar{c}_f \kappa^f \int_{\bar{n}}^{\bar{n}} \bar{r} (\bar{r} - \bar{e}) d\bar{r}] \theta^{*}$$

$$+ \frac{3}{8} \bar{c} [\int_{\bar{n}}^{1} \bar{r} (\bar{r} - \bar{e}) d\bar{r} + (\kappa^{bf} - 1) \int_{\bar{n}}^{\bar{n}} \bar{r} (\bar{r} - \bar{e}) d\bar{r}]\ \beta$$

$$+ \frac{1}{2} [\int_{\bar{n}}^{1} \bar{r}^2 (\bar{r} - \bar{e}) d\bar{r} + (\kappa^{bf} - 1) \int_{\bar{n}}^{\bar{n}} \bar{r}^2 (\bar{r} - \bar{e}) d\bar{r}]\ \theta$$

$$- \frac{1}{2} [\int_{\bar{n}}^{1} \bar{r} (\bar{r} - \bar{e}) d\bar{r} + (\kappa^{bf} - 1) \int_{\bar{n}}^{\bar{n}} \bar{r} (\bar{r} - \bar{e}) d\bar{r}]\ \lambda$$

$$+ \frac{\bar{c}_f}{8} \kappa^f \int_{\bar{n}}^{\bar{n}} \bar{r} (\bar{r} - \bar{e}) d\bar{r}\ \delta^{*}$$

$$+ \frac{c_{l\delta}}{2\,a} \int_{\bar{n}}^{\bar{n}} \bar{r}^2 (\bar{r} - \bar{e}) d\bar{r}\ \delta$$

$$+ \frac{1}{2} [\int_{\bar{n}}^{\bar{n}} \bar{r}^2 (\bar{r} - \bar{e}) d\bar{r} + \int_{\bar{n}}^{1} \bar{r}^2 (\bar{r} - \bar{e}) d\bar{r}]\ \frac{c_{l0}^{b}}{a}$$

$$+\frac{1}{2}\left[\int_{\bar{n}}^{1}\bar{r}^{2}\,(\bar{r}-\bar{e})d\bar{r}+(\kappa^{bf}-1)\int_{\bar{n}}^{\bar{n}}\bar{r}^{2}\,(\bar{r}-\bar{e})d\bar{r}\right]\theta_{0}$$

$$-\frac{1}{2}\left[\int_{\bar{n}}^{\bar{r}_1}\bar{r}^{3}\,(\bar{r}-\bar{e})d\bar{r}\right]\theta_{t1}$$

$$-\frac{1}{2}\left[\int_{\bar{n}}^{1}\bar{r}^{3}\,(\bar{r}-\bar{e})d\bar{r}+(\kappa^{bf}-1)\int_{\bar{n}}^{\bar{n}}\bar{r}^{3}\,(\bar{r}-\bar{e})d\bar{r}\right]\theta_{t2}$$

$$+\frac{\bar{r}_1}{2}\left[\int_{\bar{n}}^{1}\bar{r}^{2}\,(\bar{r}-\bar{e})d\bar{r}+(\kappa^{bf}-1)\int_{\bar{n}}^{\bar{n}}\bar{r}^{2}\,(\bar{r}-\bar{e})d\bar{r}\right](\theta_{t2}-\theta_{t1})\ \Big\}$$

(3.13)

Feathering:

$$\overline{M_\theta}^{A}=\gamma\int_{\bar{n}}^{1}\frac{\overline{M}}{a\,\bar{c}}\,d\bar{r}$$

$$=\gamma\Big\{\Big[\frac{\bar{c}^{2}}{32}(e_A+\frac{1}{4})\Big(\int_{\bar{n}}^{1}(\bar{r}-\bar{e})d\bar{r}+(\kappa^{bf}-1)\int_{\bar{r}_1}^{\bar{n}}(\bar{r}-\bar{e})d\bar{r}\Big)$$

$$-\frac{\bar{c}_f^{2}}{32}\kappa^{f}\Big(\frac{\bar{e}_f}{\zeta^{f}}+\frac{1}{4}\Big)\int_{\bar{n}}^{\bar{n}}(\bar{r}-\bar{e})d\bar{r}\Big]\beta^{**}$$

$$-\Big[\frac{\bar{c}_f^{2}}{32}\kappa^{f}\bar{c}\,\bar{e}_f\Big(\frac{\bar{e}_f}{\zeta^{f}}+\frac{1}{4}\Big)\int_{\bar{n}}^{\bar{n}}d\bar{r}\Big]\theta^{**}$$

$$+\frac{\bar{c}}{2}\bar{e}_A\Big[\int_{\bar{n}}^{1}\bar{r}\,(\bar{r}-\bar{e})d\bar{r}+(\kappa^{bf}-1)\int_{\bar{n}}^{\bar{n}}\bar{r}\,(\bar{r}-\bar{e})d\bar{r}\Big]\beta^{*}$$

$$-\frac{1}{8}\Big[\bar{c}^{2}\Big(3\,\bar{e}_A+\frac{1}{4}+\frac{\bar{e}_A}{4}+\frac{1}{16}\Big)\Big(\int_{\bar{n}}^{1}\bar{r}\,d\bar{r}+(\kappa^{bf}-1)\int_{\bar{n}}^{\bar{n}}\bar{r}\,(\bar{r}-\bar{e})d\bar{r}\Big]$$

$$+\frac{\bar{c}_f^{2}}{4}\kappa^{f}\Big(\frac{\bar{e}_f}{\zeta^{f}}+\frac{1}{4}\Big)\int_{\bar{n}}^{\bar{n}}\bar{r}\,d\bar{r}\Big]\theta^{*}$$

$$
\begin{aligned}
&- \frac{\bar{c}^2}{8}\left(3\,\bar{e}_A + \frac{1}{4}\right)\left[\int_{\bar{n}}^{1} \bar{r}\,d\bar{r} + \left(\kappa^{bf} - 1\right)\int_{\bar{n}}^{\bar{n}} \bar{r}\,d\bar{r}\right]\beta \\
&- \frac{\bar{c}}{2}\bar{e}_A\left[\int_{\bar{n}}^{1} \bar{r}^2\,d\bar{r} + \left(\kappa^{bf} - 1\right)\int_{\bar{n}}^{\bar{n}} \bar{r}^2\,d\bar{r}\right]\theta \\
&+ \frac{\bar{c}}{2}\bar{e}_A\left[\int_{\bar{n}}^{1} \bar{r}\,d\bar{r} + \left(\kappa^{bf} - 1\right)\int_{\bar{n}}^{\bar{n}} \bar{r}\,d\bar{r}\right]\lambda \\
&- \frac{\bar{c}_r^2}{32}\,\kappa^f\left(\frac{\bar{e}_f}{\zeta^f} + \frac{1}{4}\right)\int_{\bar{n}}^{\bar{n}} \bar{r}\,d\bar{r}\;\delta^* \\
&+ \frac{c_{m\delta}}{2\,a}\,\bar{c}\int_{\bar{n}}^{\bar{n}} \bar{r}^2\,d\bar{r}\;\delta \\
&+ \frac{\bar{c}}{2}\left[\int_{\bar{n}}^{\bar{n}} \bar{r}^2\,d\bar{r} + \int_{\bar{n}}^{1} \bar{r}^2\,d\bar{r}\right]\frac{c^b_{m0}}{a} \\
&- \frac{\bar{c}}{2}\bar{e}_A\left[\int_{\bar{n}}^{1} \bar{r}^2\,d\bar{r} + \left(\kappa^{bf} - 1\right)\int_{\bar{n}}^{\bar{n}} \bar{r}^2\,d\bar{r}\right]\theta_0 \\
&+ \frac{\bar{c}}{2}\bar{e}_A\left[\int_{\bar{n}}^{\bar{n}} \bar{r}^3\,d\bar{r}\right]\theta_{t1} \\
&+ \frac{\bar{c}}{2}\bar{e}_A\left[\int_{\bar{n}}^{1} \bar{r}^3\,d\bar{r} + \left(\kappa^{bf} - 1\right)\int_{\bar{n}}^{\bar{n}} \bar{r}^3\,d\bar{r}\right]\theta_{t2} \\
&- \frac{\bar{c}}{2}\bar{e}_A\,\bar{r}_1\left[\int_{\bar{n}}^{1} \bar{r}^2\,d\bar{r} + \left(\kappa^{bf} - 1\right)\int_{\bar{n}}^{\bar{n}} \bar{r}^2\,d\bar{r}\right]\left(\theta_{t2} - \theta_{t1}\right)\Big\}
\end{aligned}
\tag{3.14}
$$

To examine the importance of noncirculatory terms (the terms printed bold in Equations (3.13) and (3.14)), eigenvalue analyses were performed. For analysis of pitch horn controlled rotors, apparent mass terms can usually be neglected; Figure 12 shows that the differences in amplitude and phase for $\beta(\theta)$ with and without apparent mass terms (solid and dashed lines, respectively) are indeed insignificant (where $\beta(\theta)$ is the transfer function obtained from the ratio $\beta(\delta)/\theta(\delta)$, system I). This is not the case if servo flap deflection is the control input: Table 2 lists the roots of the transfer functions for systems I, II, and III. It is obvious that, especially in the feathering degree of freedom, neglecting of the apparent mass terms results in underestimating the system damping.

This finding is reflected in the frequency response plots of the three systems (Figures 13 to 15). The difference between the response of a model which includes apparent mass terms and one only including quasi-steady aerodynamics is more obvious for the feathering degree of freedom than for flapping. Physically, this is not surprising; the apparent mass moment terms of the flapped blade section (3.12) include apparent mass flap lift contributions multiplied by the comparably large distance between blade feathering axis and the aerodynamic center of the flap. Contribution of the blade alone can be expected to be much smaller, since the blade aerodynamic center is only 2% of the blade chord aft of the feathering axis. Thus, the effect observed can largely be attributed to the servo flap.

Comparing the frequency responses of flapping and feathering for system I with those of systems II and III leads to the conclusion that the mechanical feedbacks on the 101 Rotor were introduced in order to reduce resonance around 1/rev. It was noted in Chapter I that negative flapping

feedback reduces the aerodynamic "spring rate", corresponding to negative δ_3 angle characteristics, whereas positive feathering feedback adds a restoring aerodynamic moment to the feathering degree of freedom. In other words, the roots of flapping and feathering are moved further apart from each other, since the flapping natural frequency is reduced and the feathering natural frequency of the aerodynamically and mechanically coupled system is increased. This physical explanation is supported by results of the eigenvalue analysis (Table 2): the natural frequencies $\bar{\omega}$ in flapping and feathering of system I are indeed further apart from each other than are those of system II. Thus, by means of natural frequency separation, coupling between the two degrees of freedom is reduced, and resonance occurs less abruptly.

Based on the results of the analyses in this section, it has been decided to include mechanical feedback in the model to simulate the enhanced stability of the 101 Rotor, and to retain apparent mass terms in the aerodynamic expressions.

CHAPTER 4

DYNAMIC WAKE MODEL

4.1 General Theory

In this section, a brief sketch of the dynamic inflow theory of Peters and He is given. More detailed descriptions can be found in References [9] and [11].

For incompressible flow with small perturbations, the continuity and the momentum equation can be written in index notation:

$$q_{i,i} = 0 \tag{4.1}$$

$$q_i^* - V_\infty q_{i,\xi} = - \Phi_{,i} \tag{4.2}$$

where V_∞ is the nondimensional freestream velocity, q_i are the perturbation velocity components and q_i^*, $q_{i,\xi}$, and $q_{i,i}$ their derivatives with respect to nondimensional time, along freestream direction, and along coordinate direction, respectively. The form of (4.2) suggests separation of the perturbation pressure Φ into a part resulting from acceleration Φ^A and a part stemming from the momentum flux Φ^V, so that

$$\Phi = \Phi^A + \Phi^V \tag{4.3}$$

where

$$\Phi^V{}_{,i} = V_\infty q_{i,\xi} \tag{4.4}$$

and

$$\Phi^A{}_{,i} = -q_i^* \tag{4.5}$$

By differentiating (4.2) with respect to i and applying the continuity equation (4.1) it can be shown that both parts of the perturbation pressure must satisfy Laplace's equation and therefore resemble acceleration potentials. One solution for Laplace's equation is known as Prandtl's acceleration potential function for circular wings in the ellipsoidal coordinates ν, η, and $\bar{\psi}$

$$\Phi(\nu, \eta, \bar{\psi}, \bar{t}) = \sum_{m,n} P_n^m(\nu)\, Q_n^m(i\eta)\, [\, C_n^m(\bar{t}) \cos(m\bar{\psi}) + D_n^m(\bar{t}) \sin(m\bar{\psi}) \,] \tag{4.6}$$

using the Legendre Polynomials of the first and second kind, P_n^m and Q_n^m, and coefficients C_n^m and D_n^m. On the rotor disk, where $\eta = 0$, $\nu = \sqrt{1 - \bar{r}^2}$ and $\bar{\psi} = \psi$, and for $n + m$ odd, this function models a pressure discontinuity. The pressure difference between upper and lower side of the rotor disk ϕ is then

$$\phi(\bar{r}, \psi, \bar{t}) = -2 \sum_{m,n} P_n^m(\nu)\, Q_n^m(i0)\, [\, C_n^m(\bar{t}) \cos(m\psi) + D_n^m(\bar{t}) \sin(m\psi) \,] \tag{4.7}$$

or, written in a normalized form:

$$\phi(\bar{r},\psi,\bar{t}) = \sum_{m,n} \bar{\bar{P}}_n^m(v)\ [\tau_n^{mc}(\bar{t})\cos(m\psi) + \tau_n^{ms}(\bar{t})\sin(m\psi)] \tag{4.8}$$

where

$$\bar{\bar{P}}_n^m(v) = (-1)^m P_n^m/\rho_n^m \tag{4.9}$$

$$\tau_n^{mc} = (-1)^{m+1}\, 2\, Q_n^m(i0)\, C_n^m\, \rho_n^m \tag{4.10}$$

$$\tau_n^{ms} = (-1)^{m+1}\, 2\, Q_n^m(i0)\, D_n^m\, \rho_n^m \tag{4.11}$$

$$\rho_n^m = \sqrt{\frac{1}{2n+1}\frac{(n+m)!}{(n-m)!}} \tag{4.12}$$

If the perturbation velocity normal to the rotor disk q_z is denoted by λ, Equations (4.4) and (4.5) can be rewritten in the form

$$\lambda(\bar{r},\psi,\bar{t}) = \frac{1}{V_\infty}\int_{-\infty}^{0}\left(\frac{\partial \Phi^V}{\partial z}\right)d\xi \equiv L[\phi^V] \tag{4.13}$$

$$\lambda^* = -\left(\frac{\partial \Phi^A}{\partial \xi}\right)_{\eta=0} \equiv C[\phi^A] \tag{4.14}$$

Equation (4.7) shows that ϕ is linearly dependent on Φ. Thus, the operations on Φ^A and Φ^V in (4.13) and (4.14) can also be expressed as linear operations L and C on ϕ^A and ϕ^V. Provided that these two operators are invertible, a first order differential equation in λ can be written:

$$C^{-1}[\lambda^*] + L^{-1}[\lambda] = \phi^A + \phi^V = \phi \tag{4.15}$$

The inversion is possible if the induced velocity is expanded in terms of harmonics azimuthally and arbitrary functions radially, e.g.

$$\lambda(\bar{r},\psi,\bar{t}) = \sum_{m,n} \bar{P}_n^m(v)/v \; [\, \alpha_n^m(\bar{t})\cos(m\psi) + \beta_n^m(\bar{t})\sin(m\psi)\,] \tag{4.16}$$

introducing the inflow states α_n^m and β_n^m as coefficients of the azimuthal harmonic, m, and the radial expansion function, n. Substituting λ as in Equation (4.16) into the differential equation (4.15), premultiplying by $\bar{P}_n^m$ and $\cos(m\psi)$ and integratiing over the rotor disk yields a set of first order ordinary differential equations in α_n^m:

$$\begin{bmatrix} \ddots & & \\ & K_n^m & \\ & & \ddots \end{bmatrix} \begin{bmatrix} \ddots & & \\ & [A_{nj}^m] & \\ & & \ddots \end{bmatrix} \begin{Bmatrix} \vdots \\ \{\alpha_j^m\} \\ \vdots \end{Bmatrix}^{\cdot}$$

$$+ V_\infty \begin{bmatrix} & \vdots & \\ \cdots & [\tilde{L}_{jn}^{rmc}] & \cdots \\ & \vdots & \end{bmatrix}^{-1} \begin{bmatrix} \ddots & & \\ & [B_{nj}^r] & \\ & & \ddots \end{bmatrix} \begin{Bmatrix} \vdots \\ \{\alpha_j^m\} \\ \vdots \end{Bmatrix} = \frac{1}{2} \begin{Bmatrix} \vdots \\ \{\tau_n^{mc}\} \\ \vdots \end{Bmatrix} \tag{4.17}$$

and an equivalent equation for the β_n^m, multiplied by $\sin(m\psi)$ (replace superscript c by s). For the special choice of $\bar{P}_n^m/v$ as radial expansion functions in (4.16), the matrix elements take the form

$$K_n^m = \frac{2}{\pi} H_n^m \tag{4.18}$$

$$H_n^m = \frac{(n+m-1)!!\,(n-m-1)!!}{(n+m)!!\,(n-m)!!} \tag{4.19}$$

$$A_{nj}^{m} = \delta_{nj} \tag{4.20}$$

$$B_{nj}^{m} = \sqrt{H_j^m / H_n^m}\,(-1)^{(n+j-2m-2)/2}\sqrt{(2n+1)(2j+1)}\sum_{s=m,m+2}^{j-1} H_s^m \frac{2s+1}{(n-s)(n+s+1)} \tag{4.21}$$

The $[\widehat{L}]$ - matrix contains the interharmonic coupling terms in forward flight condition due to skewing of the wake by an angle χ (0° for axial flight, 90° for edgewise flow). For the matrix products

$$\left[\widetilde{L}^{c}\right]^{-1} \equiv \begin{bmatrix} \vdots & \\ \cdots \quad [\widehat{L}_{jn}^{rmc}] & \cdots \\ \vdots & \end{bmatrix}^{-1} \begin{bmatrix} \ddots & \\ & [B_{nj}^{r}] \\ & & \ddots \end{bmatrix} \tag{4.22}$$

$$\left[\widetilde{L}^{s}\right]^{-1} \equiv \begin{bmatrix} \vdots & \\ \cdots \quad [\widehat{L}_{jn}^{rms}] & \cdots \\ \vdots & \end{bmatrix}^{-1} \begin{bmatrix} \ddots & \\ & [B_{nj}^{r}] \\ & & \ddots \end{bmatrix} \tag{4.23}$$

the following expressions are given in Reference [11]:

$$\left[\widetilde{L}_{jn}^{0\,m}\right]^{c} = X^{m}\left[\Gamma_{jn}^{0\,m}\right] \tag{4.24}$$

$$\left[\widetilde{L}_{jn}^{r\,m}\right]^{c} = \left[X^{|m-r|} + (-1)^{l} X^{|m+r|}\right]\left[\Gamma_{jn}^{r\,m}\right] \tag{4.25}$$

$$\left[\widetilde{L}_{jn}^{r\,m}\right]^{s} = \left[X^{|m-r|} - (-1)^{l} X^{|m+r|}\right]\left[\Gamma_{jn}^{r\,m}\right] \tag{4.26}$$

where $X = \tan|\chi/2|$, $l = \min(r,m)$, and

$$\Gamma_{jn}^{rm} = \frac{(-1)^{(n+j-2r)/2}}{\sqrt{H_n^m H_j^r}} \frac{2\sqrt{(2n+1)(2j+1)}}{(j+n)(j+n+2)[(j-n)^2-1]} \quad \text{for } r+m \text{ even} \tag{4.27}$$

$$\Gamma_{jn}^{rm} = \frac{\pi}{2\sqrt{H_n^m H_j^r}} \frac{sgn(r-m)}{\sqrt{(2n+1)(2j+1)}} \quad \text{for } r+m \text{ odd}, j = n \pm 1 \tag{4.28}$$

$$\Gamma_{jn}^{rm} = 0 \quad \text{for } r+m \text{ odd}, j \neq n \pm 1 \tag{4.29}$$

The right hand side of (4.17) consists of the aerodynamic forcing functions of all rotor blades q

$$\tau_n^{0c} = \frac{1}{2\pi} \sum_q \int_{\bar{r}_i}^{1} \bar{L}_q \bar{P}_n^0(v)/v \, d\bar{r} \tag{4.30}$$

$$\tau_n^{mc} = \frac{1}{\pi} \sum_q \int_{\bar{r}_i}^{1} \bar{L}_q \bar{P}_n^m(v)/v \, d\bar{r} \cos(m\psi) \tag{4.31}$$

$$\tau_n^{ms} = \frac{1}{\pi} \sum_q \int_{\bar{r}_i}^{1} \bar{L}_q \bar{P}_n^m(v)/v \, d\bar{r} \sin(m\psi) \tag{4.32}$$

At this point, a few notes concerning partitioning and ordering of the matrices are useful. When introducing the Legendre polynomials it was mentioned that only odd combinations n+m yield a pressure discontinuity at the rotor disk and therefore a physically meaningful pressure distribution. Furthermore, the Legendre function is only defined for $n \geq m$. For a given harmonic number m, n can therefore only take on the values n = m + 1, m + 3,

m + 5, m + 7, and so on. For practical applications the expansion has to be terminated at a certain harmonic number M, which is equivalent to the highest order of $\bar{r}$ occurring. For radial expansion functions $\bar{P}_n^m(\nu)/\nu$ this means that n must not exceed N = M + 1. Under these premises, a relation between the highest power of $\bar{r}$ and the number of inflow states can be established. Table 3 reflects this ordering scheme which is expressed mathematically in Chapter 5.

The matrices are ordered in partitions, according to their harmonic number, m. Within each partition, the polynomial number, n, locates a specific element. Since these n are not consecutive numbers, they must not be mistaken for traditional row or column indices. For m = 0 all terms in the sin-partition would be zero, so m runs only from 1 to M in this part of the inflow equations. In the cos - partition m starts at 0, in Equation (4.30) with a slightly modified element.

So far, the derivation bases on the linear momentum equation (4.2); for greater generality, e.g. the hover case $V_\infty = 0$, (4.17) is slightly modified by replacing V_∞ by an equivalent mass flow parameter V

$$V = \frac{\mu^2 + (\tilde{\lambda} + \lambda_m)\,\tilde{\lambda}}{\sqrt{\mu^2 + \tilde{\lambda}^2}} \tag{4.33}$$

$$\tilde{\lambda} = \lambda_m + \lambda_f \tag{4.34}$$

where λ_m is the inflow obtained from simple momentum theory and λ_f the inflow due to freestream $V_\infty \sin\alpha$ (α being the rotor disc angle of attack). For a trimmed rotor

$$\lambda_m = \frac{1}{2}\frac{c_T}{V_T} = \sqrt{3}\,\bar{\alpha}_1^0 \tag{4.35}$$

$$V_T = \sqrt{\mu^2 + \tilde{\lambda}^2} \tag{4.36}$$

so that for axial flow ($\mu = 0$) the following trim condition can be formulated:

$$\bar{\alpha}_1^0 = -\frac{V_\infty}{2\sqrt{3}} \pm \sqrt{\left(\frac{V_\infty}{2\sqrt{3}}\right)^2 + \frac{c_T}{6}} \quad (+ \text{ for ascent, - for descent}) \tag{4.37}$$

For trim, the quantities are total values and not perturbations; in this case, the first row of $\left[\tilde{L}^c\right]^{-1}$ (corresponding to m=0, n=1) is to be multiplied by V_T, whereas all other rows have to be multiplied by V. This nonlinear version was used in the program to obtain the trim state of the model in hover; for the frequency response the linear version was applied.

4.2 Transformation into the Rotating System

Through the forcing functions τ_n^{mc} and τ_n^{ms} periodic terms enter the inflow equation (4.17). These terms can be eliminated if the reference frame rotates with the rotor. The azimuthal position of blade q is then measured as the position $\bar{t}$ of a reference blade in the nonrotating frame plus the position of blade q with respect to the reference blade in the rotating frame, $\tilde{\psi}_q$:

$$\psi_q = \bar{t} + \tilde{\psi}_q \tag{4.38}$$

where

$$\tilde{\psi}_q = \frac{2\pi}{Q} (q - 1) \tag{4.39}$$

introducing the number of blades Q. With the substitution (4.38) equation (4.16) can be rewritten in the rotating frame

$$\begin{aligned} \lambda (\bar{r}, \psi, \bar{t}) &= \sum_{m,n} \overline{P}_n^m(v)/ v \ [\alpha_n^m(\bar{t}) \cos (m\psi) + \beta_n^m(\bar{t}) \sin (m\psi)] \\ &= \sum_{m,n} \overline{P}_n^m(v)/ v \ [a_n^m(\bar{t}) \cos (m\tilde{\psi}) + b_n^m(\bar{t}) \sin (m\tilde{\psi})] \end{aligned} \tag{4.40}$$

The coefficients α_n^m, β_n^m, a_n^m and b_n^m are related by the coordinate transformations

$$\begin{Bmatrix} \alpha_n^m \\ \beta_n^m \end{Bmatrix} = \begin{bmatrix} \cos(m\bar{t}) & -\sin(m\bar{t}) \\ \sin(m\bar{t}) & \cos(m\bar{t}) \end{bmatrix} \begin{Bmatrix} a_n^m \\ b_n^m \end{Bmatrix} \tag{4.41}$$

so that

$$\begin{Bmatrix} \alpha_n^m \\ \beta_n^m \end{Bmatrix}^* = \begin{bmatrix} \cos(m\bar{\tau}) & -\sin(m\bar{\tau}) \\ \sin(m\bar{\tau}) & \cos(m\bar{\tau}) \end{bmatrix} \begin{Bmatrix} a_n^m \\ b_n^m \end{Bmatrix}^* + m \begin{bmatrix} -\sin(m\bar{\tau}) & -\cos(m\bar{\tau}) \\ \cos(m\bar{\tau}) & -\sin(m\bar{\tau}) \end{bmatrix} \begin{Bmatrix} a_n^m \\ b_n^m \end{Bmatrix} \quad (4.42)$$

Using these transformations to replace the state variables for the non-rotating coordinate system by those for the rotating frame in (4.17) , we obtain for axial flow cases (where $[\hat{L}]$ is an identity matrix)

$$\begin{bmatrix} \ddots & & & \\ & [K_n^m] & & \\ & & \ddots & \\ & & & [K_n^m] \ddots \end{bmatrix} \begin{Bmatrix} \vdots \\ \{a_j^m\} \\ \vdots \\ \{b_j^m\} \\ \vdots \end{Bmatrix}^* + \begin{bmatrix} \ddots & & \ddots & \\ & V[B_{nj}^m] & & -m[K_n^m] \ddots \\ \ddots & & \ddots & \\ & [K_n^m] & & V[B_{nj}^m] \ddots \end{bmatrix} \begin{Bmatrix} \vdots \\ \{a_j^m\} \\ \vdots \\ \{b_j^m\} \\ \vdots \end{Bmatrix} = \frac{1}{2} \begin{Bmatrix} \vdots \\ \{\tilde{\tau}_n^{mc}\} \\ \vdots \\ \{\tilde{\tau}_n^{ms}\} \\ \vdots \end{Bmatrix} \quad (4.43)$$

The forcing functions in the rotating system are then formulated as

$$\tilde{\tau}_n^{0c} = \frac{1}{2\pi} \sum_q \int_{\bar{\tau}_0}^{\bar{t}} \bar{L}_q \frac{1}{\nu} \bar{P}_n^0 \ (\nu) d\bar{\tau} \quad (4.44)$$

$$\tilde{\tau}_n^{mc} = \frac{1}{\pi} \sum_q \int_{\bar{\tau}_0}^{\bar{t}} \bar{L}_q \frac{1}{\nu} \bar{P}_n^m \ (\nu) d\bar{\tau} \ \cos(m\tilde{\psi}_q) \quad (4.45)$$

$$\tilde{\tau}_n^{ms} = \frac{1}{\pi} \sum_q \int_{\bar{\tau}_0}^{\bar{t}} \bar{L}_q \frac{1}{\nu} \bar{P}_n^m \ (\nu) d\bar{\tau} \ \sin(m\tilde{\psi}_q) \quad (4.46)$$

CHAPTER 5

COUPLED ROTOR / WAKE DYNAMICS

5.1 Complete System

Before assembling the two dynamic systems derived in Chapters 3 and 4 it is useful to recall the structure of the equations of motion. First of all, the aerodynamic forcing functions (4.44) - (4.46) are to be defined more in detail using the airload expressions of section 3.2 , which has already been done for the blade dynamic system in section 3.3. Centerpiece of the $\bar{\tau}_n^{mc}$ and $\bar{\tau}_n^{ms}$ is the integral

$$\int_{\bar{n}}^{1} \bar{L}_q \frac{1}{v} \bar{P}_n^m(v) d\bar{r} =$$

$$a\,\bar{c}\,\Big[\Big[\frac{\bar{c}}{8}\Big(\int_{\bar{n}}^{1}\frac{1}{v}\bar{P}_n^m(v)\,(\bar{r}-\bar{e})d\bar{r}+(\kappa^{bf}-1)\int_{\bar{n}}^{\bar{n}}\frac{1}{v}\bar{P}_n^m(v)\,(\bar{r}-\bar{e})d\bar{r}\Big)$$

$$\frac{\bar{c}_f}{8}\,\kappa^f\int_{\bar{n}}^{\bar{n}}\frac{1}{v}\bar{P}_n^m(v)\,(\bar{r}-\bar{e})d\bar{r}\Big]\beta_q^{**}$$

$$+\Big[\frac{\bar{c}_f}{8}\,\kappa^f\,\bar{c}\,\bar{e}_f\int_{\bar{n}}^{\bar{n}}\frac{1}{v}\bar{P}_n^m(v)d\bar{r}\Big]\theta_q^{**}$$

$$\frac{1}{2}\Big[\int_{\bar{n}}^{1}\frac{1}{v}\bar{P}_n^m(v)\,\bar{r}\,(\bar{r}-\bar{e})d\bar{r}+(\kappa^{bf}-1)\int_{\bar{n}}^{\bar{n}}\frac{1}{v}\bar{P}_n^m(v)\,\bar{r}\,(\bar{r}-\bar{e})d\bar{r}\Big]\beta_q^{*}$$

$$+\frac{1}{8}\left[4\bar{c}\left(\int_{\bar{r}}^{1}\frac{1}{\nu}\bar{P}_n^m(\nu)\,\bar{r}\,d\bar{r}+(\kappa^{bf}-1)\int_{\bar{r}}^{\bar{n}}\frac{1}{\nu}\bar{P}_n^m(\nu)\,\bar{r}\,d\bar{r}\right]\right.$$

$$\left.+\bar{c}_f\,\kappa^f\int_{\bar{r}}^{\bar{n}}\frac{1}{\nu}\bar{P}_n^m(\nu)\,\bar{r}\,d\bar{r}\right]\theta_q^*$$

$$+\frac{3}{8}\bar{c}\left[\int_{\bar{r}}^{1}\frac{1}{\nu}\bar{P}_n^m(\nu)\,\bar{r}\,d\bar{r}+(\kappa^{bf}-1)\int_{\bar{r}}^{\bar{n}}\frac{1}{\nu}\bar{P}_n^m(\nu)\,\bar{r}\,d\bar{r}\right]\beta_q$$

$$+\frac{1}{2}\left[\int_{\bar{r}}^{1}\frac{1}{\nu}\bar{P}_n^m(\nu)\,\bar{r}^2d\bar{r}+(\kappa^{bf}-1)\int_{\bar{r}}^{\bar{n}}\frac{1}{\nu}\bar{P}_n^m(\nu)\,\bar{r}^2\,d\bar{r}\right]\theta_q$$

$$-\frac{1}{2}\sum_{k,l}\left[\int_{\bar{r}}^{1}\frac{1}{\nu}\bar{P}_n^m(\nu)\,\frac{1}{\nu}\bar{P}_l^k(\nu)\,\bar{r}\,d\bar{r}+(\kappa^{bf}-1)\int_{\bar{r}}^{\bar{n}}\frac{1}{\nu}\bar{P}_n^m(\nu)\,\frac{1}{\nu}\bar{P}_l^k(\nu)\bar{r}\,d\bar{r}\right]$$

$$\left(a_l^k\cos(l\bar{\psi}_q)+b_l^k\sin(l\bar{\psi}_q)\right)$$

$$+\frac{\bar{c}_f}{8}\,\kappa^f\int_{\bar{r}}^{\bar{n}}\frac{1}{\nu}\bar{P}_n^m(\nu)\,\bar{r}\,d\bar{r}\;\delta_q^*$$

$$+\frac{c_{l\delta}}{2\,a}\int_{\bar{r}}^{\bar{n}}\frac{1}{\nu}\bar{P}_n^m(\nu)\,\bar{r}\,(\bar{r}-\bar{e})d\bar{r}\;\delta_q$$

$$+\frac{1}{2}\left[\int_{\bar{r}}^{\bar{n}}\frac{1}{\nu}\bar{P}_n^m(\nu)\,\bar{r}^2d\bar{r}+\int_{\bar{r}}^{1}\frac{1}{\nu}\bar{P}_n^m(\nu)\,\bar{r}^2d\bar{r}\right]\frac{c_{l0}^b}{a}$$

$$+\frac{1}{2}\left[\int_{\bar{r}}^{1}\frac{1}{\nu}\bar{P}_n^m(\nu)\,\bar{r}^2d\bar{r}+(\kappa^{bf}-1)\int_{\bar{r}}^{\bar{n}}\frac{1}{\nu}\bar{P}_n^m(\nu)\,\bar{r}^2d\bar{r}\right]\theta_0$$

$$-\frac{1}{2}\left[\int_{\bar{r}}^{\bar{n}}\frac{1}{\nu}\bar{P}_n^m(\nu)\,\bar{r}^3d\bar{r}\right]\theta_{t1}$$

$$-\frac{1}{2}\left[\int_{\bar{r}}^{1}\frac{1}{\nu}\bar{P}_n^m(\nu)\,\bar{r}^3d\bar{r}+(\kappa^{bf}-1)\int_{\bar{r}}^{\bar{n}}\frac{1}{\nu}\bar{P}_n^m(\nu)\,\bar{r}^3d\bar{r}\right]\theta_{t2}$$

$$+\frac{\bar{r}_1}{2}\left[\int_{\bar{r}}^{1}\frac{1}{\nu}\bar{P}_n^m(\nu)\,\bar{r}^2d\bar{r}+(\kappa^{bf}-1)\int_{\bar{r}}^{\bar{n}}\frac{1}{\nu}\bar{P}_n^m(\nu)\,\bar{r}^2d\bar{r}\right](\theta_{t2}-\theta_{t1})\;\Big] \tag{5.1}$$

where the inflow λ has been expanded according to Equation (4.40). Like the equivalent expressions (3.13) and (3.14) for the forcing functions in the blade degrees of freedom β and θ, Equation (5.1) includes both blade and inflow state variables. Expanding the right hand side of (4.43) according to (5.1), and collecting blade states in the vector $\{x\} = \{\beta_1, \theta_1, \beta_2, \theta_2, \ldots\}^T$ and inflow states

in the vector $\{y\} = \{a_1^0, a_3^0, ..., a_n^m, ..., b_n^m, \}^T$, a short hand representation of the inflow dynamic system shows the dependencies more clearly:

$$\sum_q \left([MIB]_q \{x\}_q^{**} + [CIB]_q \{x\}_q^{*} + [KIB]_q \{x\}_q \right) + [CII]_q \{y\}^{*} + \left([KII]_q + [\Delta KII]_q \right) \{y\}$$

$$= \sum_q \left([RI]_q \{u\}_q + \{cI\}_q \right) \tag{5.2}$$

The first letter in the shorthand notation for matrices and vectors used in this work stands for the category of the matrix/vector: [M] denotes a "mass" matrix, [C] a "damping" matrix, [K] a "stiffness" matrix, [R] is used for control matrices and {c} stands for a constant vector. The second letter stands for the system these matrices have been set up for, i.e. I for the inflow dynamics (like in Equation (5.2)) and B for the blade dynamic system. The third letter finally shows which generalized coordinates are to be multiplied by these matrices, again I for inflow, and B for blade states. The vector $\{u\}_q$ includes the control input and its derivatives, i.e.

$$\{u\}_q = \begin{Bmatrix} \delta_{c,q}^{**} \\ \delta_{c,q}^{*} \\ \delta_{c,q} \end{Bmatrix} \tag{5.3}$$

Accordingly, [R] includes "mass", "damping" and "stiffness" terms. It should be noted that the matrices $[MIB]_q$, $[CIB]_q$, $[KIB]_q$, $[RI]_q$ and the constant vector $\{cI\}_q$ consist of a part independent of the blade index q, multiplied by a transformation matrix including trigonometric terms from the $\tilde{\tau}_n^{mc}$ and $\tilde{\tau}_n^{ms}$, e.g.

$$[MIB]_q = [TI]_q [\tilde{MIB}] \tag{5.4}$$

where

$$[T1]_q = \begin{bmatrix} \ddots & & \\ & 1/2 & [0] \\ & & \ddots \\ & [0] & \cos(m\tilde{\psi}_q) \\ & & \ddots \\ & [0] & \sin(m\tilde{\psi}_q) \\ & & \ddots \end{bmatrix} \tag{5.5}$$

(1/2 replaces the cos-term for m=0). Hence, once computed, $[\tilde{MIB}]$ etc. can be used for all blades, or even for a rotor with the same blade geometry, but a different number of blades Q. A similar separation is possible for the matrix containing the inflow feedback coefficients in the inflow equation, $\{\Delta KII\}_q$. Here, two transformation matrices have to be utilized: [T1] contains factors stemming from the definition of the $\tau_n{}^m$ and [T2] includes the harmonics which the inflow coefficients in (4.40) are multiplied by. Then

$$[\Delta KII]_q = [T1]_q [\Delta \tilde{KII}] [T2]_q \tag{5.6}$$

using the second transformation matrix

$$[T2]_q = \begin{bmatrix} \ddots & & & \\ & 1 & [0] & [0] \\ & & \ddots & \ddots \\ & [0] & \cos(k\tilde{\psi}_q) & \sin(k\tilde{\psi}_q) \\ & & \ddots & \ddots \end{bmatrix} \tag{5.7}$$

[T2] excludes the special case m = 0, like [T1]; the harmonics index k therefore runs only from 1 to M.

Similarly to (5.2), the blade equations of motion (2.25) (system III) can be written in matrix form:

$$[MBB]\{x\}_q''+[CBB]\{x\}_q'+[KBB]\{x\}_q+[KBI]_q\{y\}=[RB]\{u\}_q+\{cB\} \tag{5.8}$$

Here, $[KBI]_q$ can be separated into a part independent of the harmonic inflow expansion and therefore of the $\tilde{\Psi}_q$, multiplied by the transformation matrix [T2];

$$[KBI]_q = [\widetilde{KBI}][T2]_q \tag{5.9}$$

The separation of common terms has been used in Chapter 6; the common elements of the matrices in (5.2) and (5.8) are listed in Appendix B. Combining rotor and inflow dynamics yields the complete system of second order ordinary differential equations in blade and inflow state variables:

$$\begin{bmatrix} \ddots & & \\ & [MBB] & \\ & & \ddots \\ \cdots & [MIB]_q & \cdots [0] \end{bmatrix} \begin{Bmatrix} \{x\}_q \\ \vdots \\ \{y\} \end{Bmatrix}'' + \begin{bmatrix} \ddots & & \\ & [CBB] & \\ & & \ddots \\ \cdots & [CIB]_q & \cdots [CII] \end{bmatrix} \begin{Bmatrix} \{x\}_q \\ \vdots \\ \{y\} \end{Bmatrix}'$$

$$+ \begin{bmatrix} \ddots & & \vdots & \\ & [KBB] & [KBI]_q & \\ & & \vdots & \ddots \\ \cdots & [KIB]_q & \cdots [KII] & \end{bmatrix} \begin{Bmatrix} \vdots \\ \{x\}_q \\ \vdots \\ \{y\} \end{Bmatrix} = \begin{bmatrix} \ddots & & \\ & [RB] & \\ & & \ddots \\ \cdots & [RIB]_q & \cdots \end{bmatrix} \begin{Bmatrix} \vdots \\ \{u\}_q \\ \vdots \end{Bmatrix} + \begin{Bmatrix} \vdots \\ \{cB\} \\ \vdots \\ \{cI\} \end{Bmatrix} \tag{5.10}$$

5.2 Reduced System for Specific Control Input

Before specifying the right hand side of (5.10) for specific control input modes, it is useful to note that the control vectors $\{u\}_q$ include time derivatives of the control input δ_c. The question occurs whether it is feasible to neglect these terms or not.

By analyzing the dimension of the elements of the control matrix (sample outputs of the matrix generation module "matgen.f" are listed in Appendix C), it is seen that the control input for inflow and blade dynamics is of the form

$$RHS = \varepsilon^3 \delta^{**} + \varepsilon^2 \delta^{*} + \delta \tag{5.11}$$

For simple harmonic control inputs, right-hand-side phase and amplitude error $\Delta\varphi$ and $\Delta\phi$ if time derivatives of δ are neglected are described by

$$\Delta\varphi = \tan^{-1}\left(\frac{\bar{\omega}\varepsilon^2}{1 - \bar{\omega}^2\varepsilon^3}\right) \tag{5.12}$$

$$\Delta\phi = \sqrt{\left(1 - \bar{\omega}^2\varepsilon^3\right)^2 + \bar{\omega}^4\varepsilon^4} - 1 \tag{5.13}$$

Evaluation of these expressions reveals that for common HHC applications (i.e., for a four bladed rotor with a maximum frequency of 5/rev), the phase error is less than 3° and the amplitude error is less than 2% (ε being equal 0.1). These numbers suggest that truncation of the control input time derivatives in this frequency range will not result in significant errors in system response. In this work, however, they remained in the equations for

completeness. For future applications in HHC though, their neglect is recommended.

Generally, the right hand side of Equation (5.10) for simple harmonic control inputs takes on the form

$$\{RHS\} = \begin{Bmatrix} \vdots \\ \{(\{RB\}_3 - \bar{\omega}_R^2\{RB\}_1) + i\,\bar{\omega}_R\{RB\}_2\}\, e^{i\varphi_q} \\ \vdots \\ \sum_{q=1}^{Q} [T1]_q \{(\{RI\}_3 - \bar{\omega}_R^2\{RI\}_1) + i\,\bar{\omega}_R\{RI\}_2\} e^{i\varphi_q} \end{Bmatrix} \tilde{\delta} \tag{5.14}$$

where $\bar{\omega}_R$ is the normalized frequency in the rotating system and $\{RIB\}_3$ e.g. denotes the third column of [RIB], i.e. the "mass" term. The following sections deal with the preparationof this expression for specified control input modes. These modes establish a certain phase relation φ between control inputs of all rotor blades, so that the control input can be written as a function of one variable - and not of four. Assuming simple harmonic control inputs δ_q and using a complex formulation of the harmonic expansion, the control input δ' for the inflow partition of (5.10) takes on the form

$$\delta' = \tilde{\delta} \sum_{q=1}^{Q} e^{-i\,m\,\bar{\varphi}_q} e^{i\varphi_q} \tag{5.15}$$

Recalling the trigonometric identity

$$\sum_{q=1}^{Q} e^{-i\,2\pi\,k\,(q-1)} = \delta_{k,j}; \qquad j = 0, \pm 1, \pm 2, \ldots \tag{5.16}$$

it appears that only for harmonics m satisfying

$$\frac{m}{Q} - \frac{\varphi_q}{2\pi\,(q-1)} = j \tag{5.17}$$

terms in the sum do not cancel each other. This means that for a specific control mode only certain frequencies excite the system, and all other harmonics can be eliminated from the inflow partition of the system matrices. This reduces their dimension significantly.

Using the rules governing the ordering of the harmonic numbers m and polynomial numbers n (section 4.1), the number of radial expansion functions S per harmonic m can be expressed in a general fashion:

$$S_m = \mathrm{Int}\left(\frac{N - m}{2}\right) + 1 \tag{5.18}$$

For a model including all harmonics, m runs from 0 to N in the cos partition and from 1 to N in the sin partition. Then the dimensions of cos, sin, and total inflow partition are

$$S^{c}_{complete} = [\ \mathrm{Int}\left(\frac{N}{2}\right) + 1]\ [\ \mathrm{Int}\left(\frac{N}{2}\right) + 1 + 2\ \mathrm{Int}\left(\frac{N+1}{2}\right) - N\ ] \tag{5.19}$$

$$S^{s}_{complete} = [\ \mathrm{Int}\left(\frac{N-1}{2}\right) + 1]\ [\ \mathrm{Int}\left(\frac{N-1}{2}\right) + 2 + 2\ \mathrm{Int}\left(\frac{N}{2}\right) - N\ ] \tag{5.20}$$

$$S^{tot}_{complete} = \frac{(N+1)(N+2)}{2} \tag{5.21}$$

In the following sections, expressions for the remaining m for different control input modes are derived and applied to Equation (5.18) in order to find the truncated matrix dimensions. All considerations are for first order input modes only.

5.2.1 Deflection of one Servo Flap only

In this case, no cancellation of harmonics takes place since no defined phase relation between the blade controls exists. On the other hand, there is only one control input, so that Equation (5.14) simplifies to

$$\{RHS\} = \begin{Bmatrix} \{RB\}_3 - \overline{\omega}_R^2 \{RB\}_1 \\ \{0\} \\ \{0\} \\ \{0\} \\ \{RI\}_3 \end{Bmatrix} + i \begin{Bmatrix} \overline{\omega}_R \{RB\}_2 \\ \{0\} \\ \{0\} \\ \{0\} \\ \overline{\omega}_R \{RI\}_2 \end{Bmatrix} \tag{5.22}$$

recalling that the second time derivative of the servo flap deflection δ does not contribute to the inflow forcing function ($\{RI\}_1 = 0$).

5.2.2 Collective HHC Flap Deflection Mode

In the collective input mode all flap deflections are in phase, i.e.

$$e^{i\varphi_{q,coll}} = 1; \quad q = 1, \ldots, Q \tag{5.23}$$

so that with Equation (5.17) the remaining harmonics are

$$m_{coll} = k\,Q; \quad k = 0,\ 1,\ 2,\ 3,\ 4,\ \ldots,\ k_{max,coll};\ k_{max,coll} = \mathrm{Int}(N/4) \tag{5.24}$$

With Equation (5.18) the truncated dimensions of the inflow matrix partition are found to be

$$S^c_{coll} = [k_{max,\,coll} + 1\,]\,[\,1 + \mathrm{Int}\left(\frac{N}{2}\right)\,] - 2 \sum_{k=0}^{k_{max,\,coll}} k \tag{5.25}$$

$$S^{s}_{coll} = k_{max,\ coll}\ [\ 1 + Int\left(\frac{N}{2}\right)] - 2 \sum_{k=0}^{k_{max,\ coll}} k \tag{5.26}$$

$$S^{tot}_{coll} = S^{c}_{coll} + S^{s}_{coll} \tag{5.27}$$

For a four bladed rotor (Q=4) summation of sin and cos terms in the inflow partition can be simplified by applying the relations

$$\sum_{q=1}^{Q} \cos(m\tilde{\psi}_q) = 4;\ m = 0, 4, 8, \ldots \tag{5.28}$$

$$\sum_{q=1}^{Q} \sin(m\tilde{\psi}_q) = 0;\ m = 0, 4, 8, \ldots \tag{5.29}$$

which eliminate the sin part of the inflow partition(note the exception m=0 in Equation (5.5), where the cos term is replaced by 1/2). It follows that

$$\{RHS\}_{coll} = \left\{\begin{Bmatrix} \vdots \\ \{RB\}_3 - \bar{\omega}_R^{\,2}\{RB\}_1 \\ \vdots \\ \{4\ \{RI\}_3 \\ \{0\} \end{Bmatrix}\right\} + i \left\{\begin{Bmatrix} \vdots \\ \bar{\omega}_R\ \{RB\}_2 \\ \vdots \\ 4\,\bar{\omega}_R\ \{RI\}_2 \\ \{0\} \end{Bmatrix}\right\} \tag{5.30}$$

5.2.3 Progressing/Regressing Flap Deflection Mode

These two modes are characterized by

$$\varphi_q = \pm 2\,\frac{q-1}{Q}\,2\pi \tag{5.31}$$

(the upper sign is to be used for the progressing mode in all equations of this section), so that with Equation (5.17)

$$m_{progr/regr} = k\,Q \;-/+\; 1;\; k = 0,\; 1,\; 2,\; \ldots,\; k_{max,\ progr/regr};$$

$$k_{max,\ progr/regr} = \mathrm{Int}\left(\frac{N+1}{2}\right) - 1 \tag{5.32}$$

Applying the relation(5.32) to Equation (5.18) yields

$$S^{c}_{progr/regr} = [k_{max,\ progr/regr} + 1]\,[1 + \mathrm{Int}\left(\frac{N-1}{2}\right)] - \sum_{k=0}^{k_{max,\ progr/regr}} k = \overset{c}{S}_{progr/regr} \tag{5.33}$$

$$S^{tot}_{progr/regr} = 2\,\overset{c}{S}_{progr/regr} \tag{5.34}$$

For a four bladed rotor all harmonics except m = 1, 3, 5, ... are eliminated. For these m

$$\sum_{q=1}^{Q} e^{i\,\varphi_{q,progr/regr}}\,\cos(m\tilde{\psi}_q) = 2;\; m = 1, 3, 5, \ldots \tag{5.35}$$

$$\sum_{q=1}^{Q} e^{i\,\varphi_{q,progr/regr}}\sin(m\tilde{\psi}_q) = \pm\, i\, 2\,(-1)^{(m+1)/2};\; m = 1, 3, 5, \ldots \tag{5.36}$$

meaning that the sin partition of the remaining harmonics of the control matrix (inflow dynamics partition) are multiplied by 2i and that the terms alter their sign with m. Result is the following structure of the right hand side:

$$\{RHS\}_{progr/regr} = \left\{\begin{Bmatrix} \{RB\}_3 - \bar{\omega}_R^{\,2}\{RB\}_1 \\ \pm\,\bar{\omega}_R\,\{RB\}_2 \\ -\{RB\}_3 + \bar{\omega}_R^{\,2}\{RB\}_1 \\ \pm(-\bar{\omega}_R\,\{RB\}_2) \\ 2\{RI\}_3 \\ 2\,\bar{\omega}_R\{(-1)^{(m+1)/2}RI\}_2 \end{Bmatrix}\right\} + i\left\{\begin{Bmatrix} \bar{\omega}_R\,\{RB\}_2 \\ \pm(-1)(\{RB\}_3 - \bar{\omega}_R^{\,2}\{RB\}_1) \\ -\,\bar{\omega}_R\,\{RB\}_2 \\ \pm(\{RB\}_3 - \bar{\omega}_R^{\,2}\{RB\}_1) \\ 2\bar{\omega}_R\{RI\}_2 \\ 2\,\{(-1)^{(m+1)/2}RI\}_3 \end{Bmatrix}\right\} \tag{5.37}$$

5.2.4 Differential HHC Flap Deflection Mode

In this case

$$\varphi_{q,\,diff} = \frac{q-1}{Q} 2\pi = \tilde{\psi}_q \tag{5.38}$$

and so

$$m_{diff} = kQ +/- Q/2; \quad k_{max,\,diff} = Int\,[(N - 2)/4] \tag{5.39}$$

For these remaining harmonics

$$S^{c}_{diff} = [k_{diff} + 1]\,[1 + Int\left(\frac{N}{2}\right)] - 2 \sum_{k=0}^{k_{max,\,diff}} k = S^{s}_{diff} \tag{5.40}$$

$$S^{tot}_{diff} = 2\, S^{c}_{diff} \tag{5.41}$$

$$\sum_{q=1}^{Q} e^{i\,\varphi_{q,diff}} \cos(m\tilde{\psi}_q) = 4;\ m = 2, 6, 10, \ldots \tag{5.42}$$

$$\sum_{q=1}^{Q} e^{i\,\varphi_{q,diff}} \sin(m\tilde{\psi}_q) = 0;\ m = 2, 6, 10, \ldots \tag{5.43}$$

and the right hand takes the following shape:

$$\{RHS\}_{progr/regr} = \begin{Bmatrix} \{RB\}_3 - \bar{\omega}_R^{\,2}\{RB\}_1 \\ -\{RB\}_3 + \bar{\omega}_R^{\,2}\{RB\}_1 \\ \{RB\}_3 - \bar{\omega}_R^{\,2}\{RB\}_1 \\ -\{RB\}_3 + \bar{\omega}_R^{\,2}\{RB\}_1 \\ \{4\{RI\}_3\} \\ \{0\} \end{Bmatrix} + i \begin{Bmatrix} \bar{\omega}_R\,\{RB\}_2 \\ -\bar{\omega}_R\,\{RB\}_2 \\ \bar{\omega}_R\,\{RB\}_2 \\ -\bar{\omega}_R\,\{RB\}_2 \\ \{4\bar{\omega}_R\{RI\}_2\} \\ \{0\} \end{Bmatrix} \tag{5.44}$$

CHAPTER 6

FREQUENCY RESPONSE

6.1 Program Layout

It was oulined in the introduction that providing a tool for further investigations on the dynamic behavior of a servo flap controlled rotor was the main objective of this research. The code is therefore programmed in several "modules", linked by a very general main program, and supported by common functions and subprograms. The code includes three modules:

- the subprogram "matgen" which calculates matrix partitions common to all blades of the system mass, damping, stiffness, and control matrix as well as of the constant vector for a specific rotor configuration and stores them in a data file denoted by the FORTRAN name "MATFILE";

- the routine "trim" which calculates the blade and inflow trim states in axial flight (including the nonlinear version of the inflow model, section 4.1) and writes the results in the file "TRIMFILE";

- and finally the module "fresp" which calculates the response of the perturbation states to harmonic inputs. The results are written in "PLOTFILE".

Both "fresp" and "trim" use a set of routines collected in the file "assemble". These routines read matrix partitions from "MATFILE", perform the necessary multiplications with trigonometric terms (i.e. transformations using the matrices [T1] and [T2]; trigonometric identities for four-bladed rotors and matrix symmetries are heeded) and locate the calculated terms in the system matrix for the control input mode chosen. The dimensions of the system matrices are calculated in the main program and given to modules and routines as a variable, so that the memory allocated for these arrays is equal to the memory needed. The idea behind this is to overcome the FORTRAN-specific disadvantage of necessary pre-specification of array dimensions. Using the modular structure, no changes in the program itself have to be made should computer memory not be sufficient. Furthermore, by assembling system matrices applying cancellation of harmonics, available memory can be utilized for including harmonics higher than those of a complete model, thus increasing accuracy. The program structure should therefore enable installation on personal computers with limited random access memory RAM.

The frequent read and write processes from and to data files could be considered as a drawback in computational efficiency. For a program run where all modules are used one after the other, storing the matrix partitions in RAM would surely be advantageous, but increases the memory needed. In any case, reading preprocessed matrix partitions from a file saves CPU-time as compared to calculating them each time the program is started. The following description of a program run illustrates its structure:

The run starts with the question if the generation of a new matrix file is desired. If this is the case, the name of the file containing the structural data

of the rotor configuration is asked for; this data file name is written in the variable "DATASET". The file "K101" e.g. (Appendix C) includes the data of the Kaman 101 Rotor. Other files of this type must have the same structure.

After input of the maximum order of radial expansion functions to be considered, "matgen" is called. This module opens "DATASET" and reads the stored geometric and aerodynamic rotor data. With this information, the matrix elements are calculated using the formulas listed in Appendix B. Integrals in the radial coordinate only have been solved analytically and defined in a set of functions collected in the routine "analy". Integrals including Legendre functions are determined in "num" using ten point Gauss integration, the Legendre polynomials being calculated by the subroutine "PNM" of He and Peters. The results are written in "MATFILE"; the name of this file is assembled of the four-letter name of "DATASET", the highest number of shape functions/harmonics used (two-digit) and the suffix "MA". For the Kaman 101 Rotor and N=8 e.g. the name of "MATFILE" is "K10108MA".

If a matrix file is already existent and no generation is required, the program jumps to another input sequence asking for the flight condition to be utilized in "trim" and "fresp", i.e. in axial flight for thrust coefficient c_T and non-dimensional rate of climb V_∞, and calculates the mass flow parameters V and V_T. The user can then decide to continue with trim calculation or frequency response.

The trimming module "trim" assembles stiffness and control matrices as well as the constant vector of Equation (5.10) from partitions stored in "MATFILE". Here, the first row of the stiffness matrix (inflow partition) is multiplied by V_T, all others by V, thus obtaining the nonlinear inflow formulation. The system is solved using LU decomposition (subroutines

"ludcmp" and "lubksb") starting with a first guess for the static control input, and then iterating until the inflow coefficient $a_1{}^0$ reaches the value obtained from simple momentum theory, Equation (4.37). Iteration is actually not necessary for axial flow conditions, but a direct calculation of the static control input would require re-ordering of the system matrix ($a_1{}^0$ is not a variable, but can be calculated directly, so that V and V_T are fixed values). For forward flight though the mass flow parameters include the unknown $a_1{}^0$; in view of future extension of the code it was therefore decided to install the iteration algorithm. The trim state vector is written in "TRIMFILE", where the name is assembled similarly to "MATFILE", with "MA" replaced by "TR".

After jumping back to the main program "flaprot", control input mode and lower frequency bound, upper frequency bound and stepwidth for the frequency response need to be specified. With this information, the module "fresp" is started.

As sketched earlier in this section, "assemble" is called and the system matrices are assembled using truncations of harmonics in the inflow partition specific to the control input mode. Each frequency step, the system is solved for real and imaginary part of the states using LU decomposition in the routines "ludcmp" and "lubksb". Amplitude and phase angle or real and imaginary part of the complex responses in flapping and feathering are stored in "PLOTFILE", which is specified by the suffix "PL" plus an index for the control mode, e.g. 1 for one servo flap deflections, 2 for collective, 3 for progressing, 4 for regressing, and 5 for the differential HHC mode.

After completion of the frequency response calculation a menu in "flaprot" enables repeated runs of each of the modules separately. Generally,

all input sequences provide default values which can be specified in the head of "flaprot", thus making input more user-friendly.

6.2 System Trim

A first check for correctness of the model is provided by calculating the trim state in hover. Main concern is the influence of the maximal order of radial shape functions N necessary to achieve convergence. With increasing N the model can approximate the actual inflow distribution more accurately. For a conventional rotor, the most interesting region in this concern is the blade tip, where three-dimensional aerodynamic effects result in lift reduction usually accounted for by the tip loss factor.

He [11] investigated convergence of time-averaged inflow for a conventional four-bladed rotor in forward flight, finding that convergence is mainly due to increase in the number of radial shape functions (N was allowed to exceed M). He concluded that four harmonics and radial shape functions up to a maximal order of eight are sufficient. A servo flapped rotor however has additionally two jumps in the two-dimensional lift and moment coefficients at the inboard and outboard edge of the servo flap. The question arises as to how many shape functions are necessary to model the three-dimensional effects associated with sheding of vorticity at these radial stations.

Table 4 shows the results for blade and inflow states as a function of N, the maximal order of radial shape functions utilized, where harmonics and radial shape functions are related as in Table 3. Data are for the Kaman 101

Main Rotor at c_T = 0.0072, equivalent to the conditions at the SH-2F helicopter with 12,500 lb gross weight (including estimated 5% fuselage download).

The simple momentum theory value for the constant inflow coefficient $a^m{}_n$, the prescribed trim value, is printed in italics, to separate it from the other variable inflow states. Values smaller than 10^{-8} were assumed to be physically meaningless and set to zero to emphasize the main tendencies. All coefficients of the sin inflow partition $b^m{}_n$ are exactly zero, and non-zero coefficients in the cos partition are found only for integer multiples of the number of blades Q, i.e. at m = 0, 4, 8, and 12, corresponding to the physical fact that in the rotating system inflow peaks can only occur at the azimuthal positions of blades

With magnitudes around 0.044 radian (or 2.52°) in flapping and about 0.13 radian (or 7.45°; at 0.75 R) in feathering, the results take on reasonable values. The servo flap control input δ_c varies between -0.012 radian and -0.035 radian (-0.69° to -2.05°) corresponding to flap deflections in the range of 0.019-0.0033 radian (1.08° ... 0.19°), reflecting the important influence of N on the trim state. In general, three-dimensional effects lead to lift loss in the outboard blade sections, or moving of the radial aerodynamic center towards the rotor hub. This directly implies reduction of the flapping angle (N = 0: 2.602° / N = 12: 2.519°) and increase of the required feathering angle (N = 0: 7.013° / N = 12: 8.112°), since lift generation is to take place further inboards, in sections of lower stagnation pressure. With increasing θ less flap deflection δ is required (N = 0: 1.076° / N = 12: 0.189°), so that the steps in the lift distribution are reduced, since in the linear aerodynamic model utilized the 2D aerodynamic coefficients are directly proportional to the flap deflection

angle. In other words, it can be expected that the trim states converge more rapidly at higher N.

Table 4 leads to the conclusion that for N = 12 the trim states have not converged yet; however, considering convergence speed, computational effort for generation of the system matrices with N larger than 12 (especially numerical calculation of integrals including products of two Legendre polynomials in the elements of partition [ΔKII]), and the fact that the trim state of the nonlinear system (see section 4.1) has no effect on the dynamic behavior of the linear variant, it was concluded that increased accuracy does not justify utilization of more than 12 radial expansion functions, and further convergence investigations have not been made.

6.3 Response to HHC Control Inputs

The response of the system to harmonic inputs between 0/rev and 8/rev in the collective, progressing, regressing, and differential mode have been investigated for N = 4, 8 and 12 and are shown in figures 16, 17, 18, and 19, respectively. The plots display phase and amplitude of the blade coordinates β and θ for a stepwidth of 0.1/rev.

At first glance, the plots look very similar. N effects the amplitude mostly at frequencies less than 1.0/rev, whereas differences in phase occur for frequencies larger than 1.0/rev. The first observation is in accordance with the considerations in preceding section, the second one reflects the fact that the dynamic order of the system is changed. The amplitude is dominated by peaks at the natural frequencies in flapping and feathering (0.9894/rev

and 1.4222/rev, from table 2) and show almost no influence of inflow dynamics in this area. Still, the peak in the flapping amplitude for the regressing mode is steeper than those of the other modes due to coupling with the first inflow mode at 1.0/rev (see section 5.2), leading to the conclusion that inflow feedback is present, but not easily observable in the plots.

A four-bladed rotor in forward flight experiences most excitations by aerodynamic forces around 4.0/rev. The frequency range 3.0/rev to 5.0/rev is therefore of special interest for HHC applications to vibration reduction (as in Reference [3]). Hence, the numerical values for amplitude and phase at these frequencies have been compared with a 'no inflow dynamics' case simulated by very high wake spacing, using an astronomical thrust coefficient c_T = 200, or equivalently a mass flow parameter V = 20. The results of this case, which has been chosen because it can be run with the program without any changes in the present source code, are listed in Tables 5 and 6.

First observation is that the flapping amplitudes in the HHC modes are generally at least 20% smaller than those of the 'no inflow' cases. Furthermore, at 3/rev in the progressing, 4/rev in the collective, and 5/rev in the regressing mode, the amplitudes are reduced by another 10% to 20%. The amplitude reductions in the HHC modes are more apparent if N = 12 radial expansion functions (and as well azimuthal harmonics) are used, whereas the amplitudes in the 'no inflow' case hardly change at all.

The general effect of amplitude reduction seems to be similar to that one described for the trim state. Increase in the highest order of radial shape functions enables more accurate modeling of radial steps in the perturbation inflow distribution in the vicinity of the oscillating servo flap, similarly to the considerations in section 6.2 (perturbations in the tip region are probably

much smaller - the influence of accuracy in the radial distribution on inflow dynamics of a conventional rotor is therefore expected to be less important).

Tables 5 and 6 show also that the relative steps in the amplitudes for neighboring frequencies are almost the same for N = 8 and N = 12. For the investigated frequencies around 4/rev, it is reasonable to assume that M = 4 harmonics are sufficient to model the observed excitation of inflow harmonics accurately. Thus, like in the trim condition, convergence is also mainly due to increase in the number of radial shape functions.

Table 7 is an equivalent comparison for blade feathering. The results in amplitude and phase are very close together, showing hardly an influence of inflow dynamics or the order of the radial expansion functions. In general, the effect of unsteady aerodynamic phenomena on this degree of freedom seems therefore to be very small, suggesting simplifications in the aerodynamic moment terms (equations (3.12) and (3.14)). The influence of inflow effects in these expressions stems mainly from the terms including quasi-steady lift multiplied by the offset of the aerodynamic center of the rotor blade with respect to the blade hinge line e_A. This value is only about 2% of the chord for the 101 Rotor (Appendix A), so that all terms including it could be neglected. A more rigorous simplification for dynamic analyses would exclude all terms except the apparent mass flap contribution (which was shown to be of significance in chapter 3.3) and a truncated quasi-steady part only including the constant inflow coefficient $a^0{}_1$.

6.3 Response to IBC Control Input (Actuation of one Flap only)

Finally, the response of blade degrees of freedom of all rotor blades to harmonic flap deflections of the reference blade q = 1 have been investigated. The results are shown in Figures 20 to 23.

Blade 1 is the only actuated blade, shows the highest flapping amplitudes, and its wake dominates therefore the inflow dynamics. If the wakes of the other blades are neglected (small flapping amplitude indicates small perturbation lift generation and so little sheding of vorticity), the wake spacing is four times the value for the HHC cases (four blades actuated), corresponding to a factor four in the mass flow parameter V. The response of the blade therefore depends less on wake feedback than in the other control input modes, as can be seen in fig. 20. The Nyquist plots in fig. 20a show the characteristic shape of transfer functions with 2 pairs of complex conjugate roots in flapping and one pair in feathering, with a total phase angle passed between 0/rev and ∞/rev of 360° and 180°, respectively. This impression is underlined by the phase plots in figures 20b and 20c: The feathering phase converges to -360° (almost independent of N), whereas the flapping phase appears to approach -540° for N = 4 (the theoretical value when inflow dynamics are excluded) and -495° for N = 12. Here, the same argumentation as in the preceding chapter is applicable: for increasing N modeling of the perturbation inflow distribution is improved, and since especially at higher frequencies the latter is dominated by the local phemonena in the vicinity of the oscillating servo flap, the influence of wake dynamics is more important under these conditions.

The responses of blades 2, 3, and 4 look absolutely different. The most significant features are sharp dents in the flapping and feathering amplitude plots around the rotor natural frequencies, where the response is almost reduced to zero. For explanation of this phenomenon, a blade oscillating freely at the flapping natural frequency should be considered. Since feathering is coupled inertially with flapping, the flapping angle follows the feathering angle with a phase shift of about 90°. The perturbation angle of attack of the blade is reduced by this motion, so that aerodynamic loads decrease. Since aerodynamic lift is the forcing function in terms of wake dynamics, it can be concluded that the excitation of the inflow states is significantly reduced under these conditions. Finally, rotor inflow dynamics couple the motion of blade 1 with the rotor degrees of freedom of the other blades, so that their response is decreased. The amplitudes can be only exactly zero for the hypothetical case in which each blade section of each blade experiences a zero perturbation angle of attack though; a practical approximation of this case could be achieved by tuning the feathering root spring rate until this condition is satisfied for e.g. the section at 0.75 R.

It is interesting to observe that a second dent in the feathering amplitude plot at an excitation slightly higher than the natural frequency in feathering is much more obvious the closer the observed blade trails the reference blade, suggesting that for the present configuration the feathering angle of the aerodynamically undamped blade 4 follows closely the azimuthal slope of the inflow distribution that it experiences when rotating behind the reference blade.

The Nyquist plots for N = 4 (figures 21a, 22a, and 23a) reflect these observations: The dents in the amplitude plots correspond to loops in the

graph, which come very close to the origin. In general, the area of these loops increases with q so that the curve approaches the origin with increasing q. In fact, the loop in the flapping plot passes the origin for an azimuthal position between those of blade 3 and 4, i.e. between $\tilde{\psi}_3$ and $\tilde{\psi}_4$. The effect of this passing of the graph through zero can be observed for blade q = 4 in fig. 23a and 23c: the second loop in the Nyquist plot for feathering (23a) approaches the origin so closely that the corresponding phase plot shows a jump of almost +180° between the desmo points 1.5/rev and 1.6/rev. In the same way, the steps in the phase plots of figures 21, 22, and 23 can be explained: depending on whether the loops approach the origin, go exactly through it, or enclose it, the phase changes in a small frequency interval by less than 180°, exactly 180° or more than 180°, respectively.

Another observation is the dependence of the response on N, detectable in phase and amplitude of flapping and feathering, apparently prohibiting reduction of N. However, the flapping response between 3/rev and 5/rev allows simplifications: the amplitudes of blades 3 and 4 in this region of interest for practical applications are almost equal for N = 4 and N = 12. Blade 2 shows an about 50% smaller amplitude for N = 12, but the amplitudes are also only about 1/3 of those of blade 4. Limitation to N = 4 therefore appears to be a reasonable and conservative simplification, which reduces the computational effort significantly. Recalling that for the control input applied no truncation of inflow harmonics by trigonometric identities can be utilized (chapter 5.2), this finding might prove very useful for future practical applications of this model.

CHAPTER 7

CONCLUSIONS AND RECOMMENDATIONS

The dynamic behavior of a servo flap controlled rotor has been investigated for different control input modes in hover, using a rigid blade rotor model, quasi-steady aerodynamics and apparent mass terms from thin airfoil theory (correlated with quasi-steady, two-dimensional experimental data) and Peters' dynamic wake formulation.

In contrast to experience with conventional rotors apparent mass terms in blade aerodynamics must not be neglected. The eigenvalues of the rigid blade model were compared with more complex approaches including elastic blade degrees of freedom, and the rigid blade modes showed acceptable correlation. Calculations of blade and inflow trim state variables in hover revealed the necessity of including more than twelve radial shape functions in the nonlinear inflow formulation for convergence.

Although through utilization of a rigid blade model no coupling of rotor and inflow modes occurs between 3/rev and 5/rev, the influence of wake dynamics in this practically interesting interval was found to be significant. The flapping amplitudes are in general approximately 20% to 25%, compared with the corresponding values obtained from calculations excluding inflow dynamics. At natural frequencies of the wake, i.e. at 3/rev for progressing, at

4/rev for collective, and at 5/rev for regressing excitation, the amplitudes drop another 10% to 20% by coupling with the inflow eigenmodes. On the other hand, hardly any wake influence is detected in the feathering response.

For the case of only one actuated and three inactive servo flaps, the transfer functions fall to very small values due to reduced inflow excitation, when the flap on the reference blade is actuated at frequencies close to rigid blade eigenvalues. The rotor response generally depends heavily on the number of radial expansion functions, but between 3 and 5/rev 4 radial polynomials appear to suffice to approximate the flapping amplitudes, which eases application of the model for IBC analyses. The blade which trails directly the actuated blade responds with about 10% of its amplitude in the vicinity of the flapping natural frequency, again underlining the impact of inflow feedback.

The rigid blade model shows the importance of including wake dynamics. In the IBC case also the effect of blade eigenvalues on inflow excitation becomes apparent. Since aerodynamic forcing functions in forward flight can occur in the range of common first elastic blade bending eigenvalues, it is suggested for further research to use an elastic blade model and to compare the results with the rigid blade findings. It is also desirable to obtain experimental data for correlation of both models.

Also, a more thorough analysis of the convergence of trim states and transfer functions with the number of radial shape functions N is necessary. It is suggested to display the radial inflow distribution for trim and azimuthal cuts of the perturbation inflow in the vicinity of the flap for the frequency response.

Main emphasis should be on extension of the model to forward flight condition though. Model and program in their present form are inappropriate for controller design. After all, HHC and IBC application in axial flight conditions is more of academic than of practical interest, since periodic airloads do not occur (if not excited by control inputs).

The FORTRAN code used for these investigations can be compiled and run on Georgia Tech's Sequent S81, but failed to be processed by compilers on other machines. The reason for rejection of the applied variable matrix dimensioning in subroutines is unknown. In view of the desired installation on personal computers, rewriting the program in C could eliminate this problem and also reduce the threat of limited available memory. To increase flexibility, the number of blades should be used as a variable rather than a fixed value, considering the possible utilization of servo flap controls for other rotor configurations. Finally, a module for eigenvalue analyses remains to be added.

APPENDIX A

KAMAN 101 ROTOR DATA

In this section structural and aerodynamic data of the 101 Rotor are listed. The values taken from Reference [14] have been converted to the nondimensional forms used in this work and correspond to the values stored in the input file "K101" of the FORTRAN code (Appendix C). Exceptions are the aspect ratio-corrected aerodynamic values used in section 3.3 (in brackets).

a	=	6.839	(5.970)
$\bar{c}$	=	0.0836	
$\bar{c}_f$	=	0.0322	
$c_{l\delta}$	=	3.372	(2.323)
$c_{l0}{}^b$	=	0.1367	(0.1158)
$c_{m\delta}$	=	-0.612	(-0.4217)
$c_{m\theta}{}^b$	=	-0.004	(-0.0032)
$c_{mo}{}^{bf}$	=	0.02	(0.017)
$\bar{e}$	=	0.0313	
$\bar{e}_A$	=	0.0211	
$\bar{e}_f$	=	0.8458	
$\bar{I}_{\beta\beta}$	=	1.0	

$$\bar{I}_{\beta\theta} = 2.469*10^{-3}$$

$$\bar{I}_{\beta\delta} = 1.384*10^{-4}$$

$$\bar{I}_{\theta\theta} = 1.023*10^{-3}$$

$$\bar{I}_{\theta\delta} = 6.368*10^{-6} = \bar{I}_{\delta\delta}$$

$$\bar{K}_{\beta} = 0.0$$

$$\bar{K}_{\theta} = 5.707*10^{-4}$$

$$\bar{K}_{c} = 0.0$$

$$\bar{r}_0 = 0.2107$$

$$\bar{r}_1 = 0.5250$$

$$\bar{r}_2 = 0.6554$$

$$\bar{r}_3 = 0.8246$$

$$\bar{S}_{\beta} = 1.4532$$

$$\bar{S}_{\theta} = 9.0779*10^{-3}$$

$$\bar{S}_{\delta} = 1.5699*10^{-4}$$

$$\gamma = 6.627 \qquad (5.28)$$

$$\theta_{t1} = 0.3924$$

$$\theta_{t2} = 0.1131$$

$$\theta_0 = 0.2314$$

$$\bar{\theta} = 0.471$$

$$\bar{\beta} = 0.0$$

$$\eta = -0.43$$

$$\zeta = 0.41$$

$$\kappa^{bf} = 0.9571 \qquad (0.9517)$$

$$\kappa^{f} = 0.4086 \qquad (0.3320)$$

$$\zeta^{f} = 0.3852$$

APPENDIX B

ELEMENTS OF SYSTEM SUBMATRICES

This list provides an overview of the common matrices (blade index independend) occurring in the blade and inflow dynamic system equations of Chapter 5. Inflow dynamics are set up in the rotating system, for axial flow conditions. The following expressions have been utilized in the subroutine "matgen" of the FORTRAN code, Appendix C. In general, the first subscript denotes the row number, and the second number the column index (or the row/column partitioning in n and m).

Blade Dynamics:

Mass matrix, blade states:

$$MBB_{11} = 1 - \eta\,\bar{I}_{\beta\delta} + \gamma\,\{\frac{\bar{c}}{8}[\int_{\bar{r}_0}^{1}(\bar{r}-\bar{e})^2 d\bar{r} + (\kappa^{bf}-1)\int_{\bar{r}_2}^{\bar{r}_3}(\bar{r}-\bar{e})^2 d\bar{r}] + \frac{\bar{c}_f}{8}\kappa^f\int_{\bar{r}_2}^{\bar{r}_3}(\bar{r}-\bar{e})^2 d\bar{r}\,\}$$

$$MBB_{12} = -\,\bar{I}_{\beta\theta} - \zeta\,\bar{I}_{\beta\delta} - \gamma\,\{\frac{\bar{c}_f}{8}\kappa^f\bar{c}\,\bar{e}_f\int_{\bar{r}_2}^{\bar{r}_3}(\bar{r}-\bar{e})d\bar{r}\,\}$$

$$MBB_{21} = -\,\bar{I}_{\beta\theta} + \eta\,\bar{I}_{\theta\delta} - \gamma\,\{\frac{\bar{c}^2}{32}(\bar{e}_A+\frac{1}{4})[\int_{\bar{r}_0}^{1}(\bar{r}-\bar{e})d\bar{r} + (\kappa^{bf}-1)\int_{\bar{r}_2}^{\bar{r}_3}(\bar{r}-\bar{e})d\bar{r}]$$
$$+ \frac{\bar{c}_f^2}{32}\kappa^f\bar{c}\,\bar{e}_f(\frac{\bar{e}_f}{\zeta_f}+\frac{1}{4})\int_{\bar{r}_2}^{\bar{r}_3}(\bar{r}-\bar{e})d\bar{r}\,\}$$

$$MBB_{22} = \bar{I}_{\beta\theta} + \zeta\,\bar{I}_{\theta\delta} - \gamma\,\{\frac{\bar{c}_f^2}{32}\kappa^f\bar{c}\,\bar{e}_f(\frac{\bar{e}_f}{\zeta_f}+\frac{1}{4})\int_{\bar{r}_2}^{\bar{r}_3}d\bar{r}\,\}$$

Damping matrix, blade states:

$$CBB_{11} = \gamma \{\frac{1}{2}[\int_{\bar{r}_0}^{1} \bar{r}\,(\bar{r}-\bar{e})^2 d\bar{r} + (\kappa^{bf}-1)\int_{\bar{r}_2}^{\bar{r}_3} \bar{r}\,(\bar{r}-\bar{e})^2 d\bar{r}] - \eta\,\frac{\bar{c}_f}{8}\kappa^f \int_{\bar{r}_2}^{\bar{r}_3} \bar{r}\,(\bar{r}-\bar{e})d\bar{r}\}$$

$$CBB_{12} = -\gamma\{\frac{\bar{c}}{2}[\int_{\bar{r}_0}^{1} \bar{r}\,(\bar{r}-\bar{e})d\bar{r} + (\kappa^{bf}-1)\int_{\bar{r}_2}^{\bar{r}_3} \bar{r}\,(\bar{r}-\bar{e})d\bar{r}] + \frac{\bar{c}_f}{8}\kappa^f(1+\zeta)\int_{\bar{r}_2}^{\bar{r}_3} \bar{r}\,(\bar{r}-\bar{e})d\bar{r}\}$$

$$CBB_{21} = -\gamma\{\frac{\bar{e}_A\bar{c}}{2}[\int_{\bar{r}_0}^{1} \bar{r}\,(\bar{r}-\bar{e})d\bar{r} + (\kappa^{bf}-1)\int_{\bar{r}_2}^{\bar{r}_3} \bar{r}\,(\bar{r}-\bar{e})d\bar{r}] - \eta\,\frac{\bar{c}_f^2}{32}\kappa^f(\frac{\bar{e}_f}{\zeta_f}+\frac{1}{4})\int_{\bar{r}_2}^{\bar{r}_3} \bar{r}\,d\bar{r}\}$$

$$CBB_{22} = \gamma\{\frac{\bar{c}^2}{8}(3\bar{e}_A+\frac{1}{4})[\int_{\bar{r}_0}^{1} \bar{r}\,d\bar{r} + (\kappa^{bf}-1)\int_{\bar{r}_2}^{\bar{r}_3} \bar{r}\,d\bar{r}]$$
$$+\frac{1}{32}[\bar{c}^2\kappa^{bf}(\bar{e}_A+\frac{1}{4}) + (\bar{c}^2\kappa^{bf}+\bar{c}_f^2\kappa^f(1+\zeta))\,(\frac{\bar{e}_f}{\zeta_f}+\frac{1}{4})\int_{\bar{r}_2}^{\bar{r}_3} \bar{r}\,d\bar{r}]\}$$

Stiffness matrix, blade states:

$$KBB_{11} = \bar{K}_{\beta\beta} - \eta\,\bar{K}_{\beta\delta} - \gamma\{\frac{3}{8}\bar{c}[\int_{\bar{r}_0}^{1} \bar{r}\,(\bar{r}-\bar{e})d\bar{r} + (\kappa^{bf}-1)\int_{\bar{r}_2}^{\bar{r}_3} \bar{r}\,(\bar{r}-\bar{e})d\bar{r}]$$
$$+\eta\,\frac{c_{l\delta}}{2a}\int_{\bar{r}_2}^{\bar{r}_3} \bar{r}^2(\bar{r}-\bar{e})d\bar{r}\}$$

$$KBB_{12} = -\bar{K}_{\beta\theta} - \zeta\,\bar{K}_{\beta\delta} - \gamma\{\frac{1}{2}[\int_{\bar{r}_0}^{1} \bar{r}^2(\bar{r}-\bar{e})d\bar{r} + (\kappa^{bf}-1)\int_{\bar{r}_2}^{\bar{r}_3} \bar{r}^2(\bar{r}-\bar{e})d\bar{r}]$$
$$+\zeta\,\frac{c_{l\delta}}{2a}\int_{\bar{r}_2}^{\bar{r}_3} \bar{r}^2(\bar{r}-\bar{e})d\bar{r}]$$

$$KBB_{21} = -\bar{K}_{\beta\theta} + \eta\,\bar{K}_{\theta\delta} + \gamma\{\frac{\bar{c}^2}{8}(3\bar{e}_A+\frac{1}{4})[\int_{\bar{r}_0}^{1} \bar{r}\,d\bar{r} + (\kappa^{bf}-1)\int_{\bar{r}_2}^{\bar{r}_3} \bar{r}\,d\bar{r}]$$
$$+\eta\,\frac{c_{m\delta}}{2a}\bar{c}\int_{\bar{r}_2}^{\bar{r}_3} \bar{r}^2 d\bar{r}\}$$

$$KBB_{22} = \bar{K}_{\beta\theta} + \zeta\,\bar{K}_{\theta\delta} + \gamma\{\frac{1}{2}\bar{c}\,\bar{e}_A[\int_{\bar{r}_0}^{1} \bar{r}^2 d\bar{r} + (\kappa^{bf}-1)\int_{\bar{r}_2}^{\bar{r}_3} \bar{r}^2 d\bar{r}]$$
$$+\zeta\,\frac{c_{m\delta}}{2a}\bar{c}\int_{\bar{r}_2}^{\bar{r}_3} \bar{r}^2 d\bar{r}\}$$

Stiffness matrix, inflow states:

$$\widetilde{KBI}^{m}_{n1} = \frac{\gamma}{2}\left[\int_{\bar{r}_n}^{1} \frac{1}{\nu} P_n^m(\nu)\,\bar{r}\,(\bar{r}-\bar{e})d\bar{r} + (\kappa^{bf}-1)\int_{\bar{r}_n}^{\bar{r}_n} \frac{1}{\nu} P_n^m(\nu)\,\bar{r}\,(\bar{r}-\bar{e})d\bar{r}\right]$$

$$\widetilde{KBI}^{m}_{n2} = \frac{\gamma}{2}\overline{ce}_A\left[\int_{\bar{r}_n}^{1} \frac{1}{\nu} P_n^m(\nu)\,\bar{r}\,d\bar{r} + (\kappa^{bf}-1)\int_{\bar{r}_n}^{\bar{r}_n} \frac{1}{\nu} P_n^m(\nu)\,\bar{r}\,d\bar{r}\right]$$

Control matrix:

$$RB_{11} = \bar{I}_{\beta\delta}$$

$$RB_{12} = \frac{\gamma}{8}\bar{c}_f\kappa^f\int_{\bar{r}_2}^{\bar{r}_3} \bar{r}\,(\bar{r}-\bar{e})d\bar{r}$$

$$RB_{13} = \bar{K}_{\beta\delta} + \gamma\frac{c_{l\delta}}{2a}\int_{\bar{r}_2}^{\bar{r}_3} \bar{r}^2\,(\bar{r}-\bar{e})d\bar{r}$$

$$RB_{21} = -\bar{I}_{\theta\delta}$$

$$RB_{12} = -\frac{\gamma}{32}\bar{c}_f^2\kappa^f\left(\frac{\bar{c}_f}{\zeta_f}+\frac{1}{4}\right)\int_{\bar{r}_2}^{\bar{r}_3} \bar{r}\,d\bar{r}$$

$$RB_{13} = -\bar{K}_{\theta\delta} + \gamma\frac{c_{m\delta}}{2a}\bar{c}\int_{\bar{r}_2}^{\bar{r}_3} \bar{r}^2 d\bar{r}$$

Constant vector:

$$\begin{aligned} cB_1 = \gamma\{ & \frac{1}{2}\left[\int_{\bar{r}_n}^{\bar{r}_n} \bar{r}^2\,(\bar{r}-\bar{e})d\bar{r} + \int_{\bar{r}_n}^{1} \bar{r}^2\,(\bar{r}-\bar{e})d\bar{r}\right]\frac{c^b_{l0}}{a} \\ & + \frac{1}{2}\left[\int_{\bar{r}_n}^{1} \bar{r}^2\,(\bar{r}-\bar{e})d\bar{r} + (\kappa^{bf}-1)\int_{\bar{r}_n}^{\bar{r}_n} \bar{r}^2\,(\bar{r}-\bar{e})d\bar{r}\right]\theta_0 \\ & - \frac{1}{2}\left[\int_{\bar{r}_n}^{\bar{r}_n} \bar{r}^3\,(\bar{r}-\bar{e})d\bar{r}\right]\theta_{t1} \\ & - \frac{1}{2}\left[\int_{\bar{r}_n}^{1} \bar{r}^3\,(\bar{r}-\bar{e})d\bar{r} + (\kappa^{bf}-1)\int_{\bar{r}_n}^{\bar{r}_n} \bar{r}^3\,(\bar{r}-\bar{e})d\bar{r}\right]\theta_{t2} \\ & + \frac{\bar{r}_1}{2}\left[\int_{\bar{r}_n}^{1} \bar{r}^2\,(\bar{r}-\bar{e})d\bar{r} + (\kappa^{bf}-1)\int_{\bar{r}_n}^{\bar{r}_n} \bar{r}^2\,(\bar{r}-\bar{e})d\bar{r}\right](\theta_{t2}-\theta_{t1})\} + \bar{K}_\beta\bar{\beta} \end{aligned}$$

$$cB_2 = \gamma \{ \frac{\bar{c}}{2} [\int_{\bar{n}}^{\bar{n}} \bar{r}^2 \, d\bar{r} + \int_{\bar{n}}^{1} \bar{r}^2 \, d\bar{r}] \frac{c_{m_0}^b}{a}$$
$$- \frac{\bar{c}}{2} \bar{e}_A [\int_{\bar{n}}^{1} \bar{r}^2 \, d\bar{r} + (\kappa^{bf} - 1) \int_{\bar{n}}^{\bar{n}} \bar{r}^2 \, d\bar{r}] \theta_0$$
$$+ \frac{\bar{c}}{2} \bar{e}_A [\int_{\bar{n}}^{\bar{n}} \bar{r}^3 \, d\bar{r}] \theta_{t1}$$
$$+ \frac{\bar{c}}{2} \bar{e}_A [\int_{\bar{n}}^{1} \bar{r}^3 \, d\bar{r} + (\kappa^{bf} - 1) \int_{\bar{n}}^{\bar{n}} \bar{r}^3 \, d\bar{r}] \theta_{t2}$$
$$- \frac{\bar{c}}{2} \bar{e}_A \bar{r}_1 [\int_{\bar{n}}^{1} \bar{r}^2 \, d\bar{r} + (\kappa^{bf} - 1) \int_{\bar{n}}^{\bar{n}} \bar{r}^2 \, d\bar{r}] (\theta_{t2} - \theta_{t1}) \} + \bar{K}_\theta \bar{\theta}$$

Inflow Dynamics:

Mass matrix, blade states:

$$\tilde{MIB}_{n1}^{m} = \frac{a\bar{c}}{16\pi} [\bar{c} (\int_{\bar{n}}^{1} \frac{1}{\nu} \bar{P}_n^m(\nu) (\bar{r} - \bar{e}) d\bar{r} + (\kappa^{bf} - 1) \int_{\bar{n}}^{\bar{n}} \frac{1}{\nu} \bar{P}_n^m(\nu) (\bar{r} - \bar{e}) d\bar{r})$$
$$+ \bar{c}_f \kappa^f \int_{\bar{n}}^{\bar{n}} \frac{1}{\nu} \bar{P}_n^m(\nu) (\bar{r} - \bar{e}) d\bar{r}]$$

$$\tilde{MIB}_{n2}^{m} = \frac{a\bar{c}}{16\pi} [\bar{c}_f \kappa^f \bar{c} \bar{e}_f \int_{\bar{n}}^{\bar{n}} \frac{1}{\nu} \bar{P}_n^m(\nu) d\bar{r}]$$

Damping matrix, blade states:

$$\tilde{CIB}_{n1}^{m} = \frac{a\bar{c}}{4\pi} [\int_{\bar{n}}^{1} \frac{1}{\nu} \bar{P}_n^m(\nu) (\bar{r} - \bar{e}) d\bar{r} + (\kappa^{bf} - 1) \int_{\bar{n}}^{\bar{n}} \frac{1}{\nu} \bar{P}_n^m(\nu) (\bar{r} - \bar{e}) d\bar{r}$$
$$- \eta \frac{1}{4} \bar{c}_f \kappa^f \int_{\bar{n}}^{\bar{n}} \frac{1}{\nu} \bar{P}_n^m(\nu) d\bar{r}]$$

$$\tilde{CIB}_{n2}^{m} = - \frac{a\bar{c}}{4\pi} [\bar{c} (\int_{\bar{n}}^{1} \frac{1}{\nu} \bar{P}_n^m(\nu) \bar{r} d\bar{r} + (\kappa^{bf} - 1) \int_{\bar{n}}^{\bar{n}} \frac{1}{\nu} \bar{P}_n^m(\nu) \bar{r} \, d\bar{r})$$
$$+ (\zeta + 1) \frac{1}{4} \bar{c}_f \kappa^f \int_{\bar{n}}^{\bar{n}} \frac{1}{\nu} \bar{P}_n^m(\nu) \bar{r} \, d\bar{r}]$$

Stiffness matrix, blade states:

$$\tilde{KIB}^{m}_{n1} = \frac{\bar{a}\bar{c}}{4\pi}\left[\, 3\bar{c}\int_{\bar{r}_n}^{1}\frac{1}{v}\bar{P}^{m}_{n}(v)\,\bar{r}\,d\bar{r} + (\kappa^{bf}-1)\int_{\bar{r}_n}^{\bar{r}_n}\frac{1}{v}\bar{P}^{m}_{n}(v)\,\bar{r}\,d\bar{r} + \eta\frac{c_{l\delta}}{a}\int_{\bar{r}_n}^{\bar{r}_n}\frac{1}{v}\bar{P}^{m}_{n}\,\bar{r}^{2}\,(v)d\bar{r}\,\right]$$

$$\tilde{KIB}^{m}_{n2} = -\frac{\bar{a}\bar{c}}{4\pi}\left[\int_{\bar{r}_n}^{1}\frac{1}{v}\bar{P}^{m}_{n}(v)\,\bar{r}^{2}d\bar{r} + (\kappa^{bf}-1)\int_{\bar{r}_n}^{\bar{r}_n}\frac{1}{v}\bar{P}^{m}_{n}(v)\,\bar{r}^{2}d\bar{r} + \zeta\frac{c_{l\delta}}{a}\int_{\bar{r}_n}^{\bar{r}_n}\frac{1}{v}\bar{P}^{m}_{n}(v)\bar{r}^{2}d\bar{r}\,\right]$$

Damping matrix, inflow states:

$$[CII] = \begin{bmatrix} \ddots & & \\ & K^{m}_{n} & \\ & & \ddots \\ & & & K^{m}_{n} & \\ & & & & \ddots \end{bmatrix}$$

K^{m}_{n} see equations (4.18)/(4.19)

Stiffness matrix, inflow states:

$$[KII] = \begin{bmatrix} \ddots & & \\ & V\left[B^{m}_{jn}\right] & -mK^{m}_{n} \\ & & \ddots \\ & mK^{m}_{n} & V\left[B^{m}_{jn}\right] \\ & & \ddots \end{bmatrix}$$

$\left[B^{m}_{jn}\right]$ see equation (4.19)/(4.21)

Inflow states feedback:

$$\Delta\tilde{KII}^{k\,m}_{l\,n} = \frac{\bar{a}\bar{c}}{4\pi}\left[\int_{\bar{r}_0}^{1}\frac{\bar{P}^{k}_{l}\,\bar{P}^{m}_{n}}{v^{2}}\,\bar{r}\,d\bar{r} + (\kappa^{bf}-1)\int_{\bar{r}_2}^{\bar{r}_3}\frac{\bar{P}^{k}_{l}\,\bar{P}^{m}_{n}}{v^{2}}\,\bar{r}\,d\bar{r}\,\right]$$

Control matrix:

$$\widetilde{RI}^{m}_{n\,1} = 0$$

$$\widetilde{RI}^{m}_{n\,2} = \frac{a\bar{c}}{16\pi}\,\bar{c}_f\,\kappa^f \int_{\bar{r}_2}^{\bar{r}_3} \frac{1}{\nu}\,\overline{P}^{m}_{n}(\nu)\,\bar{r}\,d\bar{r}$$

$$\widetilde{RI}^{m}_{n\,3} = \frac{a\bar{c}}{4\pi}\,\frac{c_{l\delta}}{a} \int_{\bar{r}_2}^{\bar{r}_3} \frac{1}{\nu}\,\overline{P}^{m}_{n}(\nu)\,\bar{r}^2\,d\bar{r}$$

Constant vector:

$$\begin{aligned}
\widetilde{cI}^{m}_{n} = a\bar{c}\,\{ & \frac{1}{2}\,[\int_{\bar{n}}^{\bar{n}} \frac{1}{\nu}\,\overline{P}^{m}_{n}(\nu)\;\bar{r}^{\,2}d\bar{r} + \int_{\bar{n}}^{1} \frac{1}{\nu}\,\overline{P}^{m}_{n}(\nu)\;\bar{r}^{\,2}d\bar{r}]\;\frac{c^{b}_{l0}}{a} \\
& + \frac{1}{2}\,[\int_{\bar{n}}^{1} \frac{1}{\nu}\,\overline{P}^{m}_{n}(\nu)\;\bar{r}^{\,2}d\bar{r} + (\kappa^{bf} - 1)\int_{\bar{n}}^{\bar{n}} \frac{1}{\nu}\,\overline{P}^{m}_{n}(\nu)\;\bar{r}^{\,2}d\bar{r}]\;\theta_0 \\
& - \frac{1}{2}\,[\int_{\bar{n}}^{\bar{n}} \frac{1}{\nu}\,\overline{P}^{m}_{n}(\nu)\;\bar{r}^{\,3}d\bar{r}]\;\theta_{t1} \\
& - \frac{1}{2}\,[\int_{\bar{n}}^{1} \frac{1}{\nu}\,\overline{P}^{m}_{n}(\nu)\;\bar{r}^{\,3}d\bar{r} + (\kappa^{bf} - 1)\int_{\bar{n}}^{\bar{n}} \frac{1}{\nu}\,\overline{P}^{m}_{n}(\nu)\;\bar{r}^{\,3}d\bar{r}]\;\theta_{t2} \\
& + \frac{\bar{r}_1}{2}\,[\int_{\bar{n}}^{1} \frac{1}{\nu}\,\overline{P}^{m}_{n}(\nu)\;\bar{r}^{\,2}d\bar{r} + (\kappa^{bf} - 1)\int_{\bar{n}}^{\bar{n}} \frac{1}{\nu}\,\overline{P}^{m}_{n}(\nu)\;\bar{r}^{\,2}d\bar{r}]\;(\theta_{t2} - \theta_{t1})\;\}
\end{aligned}$$

APPENDIX C

FORTRAN CODE

The following FORTRAN code has been used for all calculations in this research. The source code was successfully compiled by the standard DYNIX 'fortran'-compiler on Georgia Tech's Sequent S81 mainframe computer and the program run on this machine.

In this form, the program allows flexible dimensioning of matrices in the subroutines, which is a desirable characteristic for investigation of the model performance. However, only the compiler on the Sequent accepted the present structure, although only standard FORTRAN 77 features have been applied. Compilers on Sun workstations and Macintosh personal computers required matrix dimensions as parameters rather than as a variable given by the main program. On the other hand, sequences of the code including the same process for matrix dimensioning in sub-subroutines (as the LU-decomposition) were compiled without problems. The reason for this failure could not be found.

For further application though the flexibility in the matrix dimensioning might not be an absolutely necessary feature; the findings of Chapter 6 suggest that e.g. for the Kaman 101 Rotor four radial expansion

functions (in FORTRAN notation NCAP=4) already show acceptable convergence in the frequency range most interesting for HHC applications (3.0 ... 5.0/rev). As a part of a controller design package e.g., this value (and so the matrix dimensions) could therefore be fixed, and the code could be split into its modules, discarding the connecting main program 'flaprot'.

For an eigenvalue analysis the 'body' of 'fresp.f' can be used without changes for assembly the system matrices (i.e. the sequence calling the routines in 'assemble.f'). Extension of the present code to modal analyses then only requires combination of the existing matrix assembly sequence with a standard eigenvalue analysis (and a corresponding section in 'flaprot.f' calling the new module, if desired).

The FORTRAN codes are ordered the way they were collected in files; each module is followed by a sample output file.

Main Program:

File 'flaprot.f'

```
      PROGRAM FLAPROT
C*****************************************************************
C*                                                               *
C*                                                               *
C*    MAIN PROGRAM FOR SIMULATION OF A SERVO FLAP CONTROLLED     *
C*    ROTOR WITH FOUR RIGID BLADES                               *
C*                                                               *
C*                                                               *
C*    FEATURES:                                                  *
C*                                                               *
C*    -DYNAMICS OF ROTOR BLADES, RIGID IN BENDING AND TORSION,   *
C*          AND SERVO FLAP; INCLUDES MECHANICAL FEEDBACK OF      *
C*          FLAPPING AND FEATHERING ANGLE IN SERVO FLAP DEFLECTION *
C*          (AS UTILIZED AT THE KAMAN 101 ROTOR)                 *
C*    -QUASI-STEADY AERODYNAMICS FROM THIN AIRFOIL THEORY        *
C*          (REF.:JOHNSON)                                       *
C*          CORRECTED BY USE OF AERODYNAMIC COEFFICIENTS TAKEN   *
C*          FROM  STEADY 2D TEST DATA                            *
C*          (BLADE AND BLADE/SERVO FLAP SECTION)                 *
C*    -APPARENT MASS TERMS FROM THIN AIRFOIL THEORY              *
C*          (LIFT CURVE SLOPE FROM STEADY 2D TEST DATA)          *
C*    -UNSTEADY AERODYNAMICS / WAKE DYNAMICS FROM PETER'S        *
C*          STATE-SPACE DYNAMIC WAKE MODEL                       *
C*                                                               *
C*                                                               *
C*    DATA FILES GENERATED:                                      *
C*                                                               *
C*    "MATFILE"  -  COMMON MATRIX PARTITIONS,                    *
C*                  NOT DEPENDENT ON MODEL TYPE                  *
C*                  SUFFIX "MA"                                  *
C*    "TRIMFILE" -  TRIM STATE                                   *
C*                SUFFIX "TR"                                    *
C*    "PLOTFILE"  -  FREQUENCY RESPONSE OF STATE VARIABLES       *
C*                SUFFIX "PL"                                    *
C*                                                               *
C*****************************************************************
      INTEGER MODEL,M0,DM,KMAX,NCAP
      INTEGER  DIM,DIMIN,DIMINC,DIMINS,DIMINSM,DIMINM
      INTEGER L,K,KSUM,SWITCH
      REAL V,VT,VINF,CT,SMT
      REAL  NRMIN,NRMAX,DNR
      CHARACTER*1 CHOICE
      CHARACTER*4 DATASET
      CHARACTER*8 MATFILE
C     DEFAULT SETTINGS
```

```
      MATFILE='K10104MA'
      CT=0.0072
      VINF=0.
      NRMIN=0.
      NRMAX=8.
      DNR=0.1
      SWITCH=1
C
      PRINT*
      PRINT*
      PRINT*
      PRINT*
      PRINT*
      PRINT*
      PRINT*
      PRINT*
      PRINT*
      PRINT*
      PRINT*
      PRINT*,'    ****************************************,'
     $'************************'
      PRINT*,'    *                                       ',
     $'  *'
      PRINT*,'    *              PROGRAM FLAPROT          ',
     $'  *'
      PRINT*,'    *                                       ',
     $'  *'
      PRINT*,'    *           AEROELASTIC SIMULATION      ',
     $'  *'
      PRINT*,'    *                  OF A                 ',
     $'  *'
      PRINT*,'    *       SERVO FLAP CONTROLLED ROTOR     ',
     $'  *'
      PRINT*,'    *          WITH FOUR RIGID BLADES       ',
     $'  *'
      PRINT*,'    *                                       ',
     $'  *'
      PRINT*,'    *            VERSION 1.7, DEC 04, 1990  ',
     $'  *'
      PRINT*,'    *  (USING LU DECOMPOSITION & SYMMETRIES) ',
     $'  *'
      PRINT*,'    *              MARTIN STETTNER          ',
     $'  *'
      PRINT*,'    *                                       ',
     $'  *'
      PRINT*,'    ****************************************',
     $'*******'
      PRINT*
      PRINT*
      PRINT*
      PRINT*
      PRINT*
```

```
      PRINT*
      PRINT*
      PRINT*
      PRINT*,'TASK: GENERATION OF NEW MATRIX FILE (Y/y, DEF=NO)?'
      READ(*,*)CHOICE
      IF(CHOICE.NE.'Y'.AND.CHOICE.NE.'y')GOTO 100
      PRINT*
C
C     GENERATION OF COMMON MATRIX PARTITIONS
C
 1    CONTINUE
      PRINT*,'GEOMETRIC/AERODYNAMIC DATA FILE (A4)?'
      READ(*,*) DATASET
      PRINT*,'MAXIMAL ORDER OF RADIAL EXPANSION FUNCTIONS (MAX=20)?'
      READ(*,*)  NCAP
      L=(NCAP-1)/2+1
      DIMINS=L*L+(2*INT(NCAP/2)-NCAP+1)*L
      DIMIN=(NCAP+1)*(NCAP+2)/2
      DIMINC=DIMIN-DIMINS
      CALL MATGEN(DATASET, NCAP, DIMIN, DIMINC)
      PRINT*
      PRINT*,'CONTINUE WITH TRIM AND/OR FREQUENCY RESPONSE (N/n)?'
      READ(*,*) CHOICE
      IF (CHOICE.EQ.'N'.OR.CHOICE.EQ.'n') STOP
C
 100  CONTINUE
      PRINT*,'USE DEFAULT MATRIX FILE ', MATFILE,' (Y/y)?'
      READ(*,*) CHOICE
      IF (CHOICE.EQ.'Y'.OR.CHOICE.EQ.'y') GOTO 102
 101  CONTINUE
      PRINT*,'MATRIX FILE TO BE USED FOR FURTHER CALCULATIONS (A8)?'
      READ(*,*)  MATFILE
      IF(MATFILE(7:).NE.'MA')THEN
        PRINT*,'INPUT ERROR: FILE NAME MUST END WITH "MA"!'
       GOTO 101
      ELSE
      END IF
 102  CONTINUE
      NCAP=10*(  ICHAR(MATFILE(5:))-48  )+ICHAR(MATFILE(6:))-48
      L=(NCAP-1)/2+1
      DIMINS=L*L+(2*INT(NCAP/2)-NCAP+1)*L
      DIMIN=(NCAP+1)*(NCAP+2)/2
      DIMINC=DIMIN-DIMINS
      WRITE(*,500) CT, VINF
      READ(*,*) CHOICE
      IF (CHOICE.EQ.'Y'.OR.CHOICE.EQ.'y') GOTO 103
      PRINT*,'THRUST COEFFICIENT CT?'
      READ(*,*) CT
      PRINT*,'RATE OF CLIMB?'
      READ(*,*)  VINF
 103  CONTINUE
      IF(VINF.GE.0)THEN
```

```
          SMT=(-VINF/2.+SQRT(VINF/2.*VINF/2.+CT/2.))/SQRT(3.)
        ELSE
          SMT=(-VINF/2.-SQRT(VINF/2.*VINF/2.+CT/2.))/SQRT(3.)
        END IF
        VT=VINF+SQRT(3.)*SMT
        V=VINF+2.*SQRT(3.)*SMT
        PRINT*
        PRINT*,'TASK: CALCULATION OF TRIM STATE (Y/y, DEF=NO)?'
        READ(*,*)CHOICE
        IF(CHOICE.NE.'Y'.AND.CHOICE.NE.'y')GOTO 200
        PRINT*
C
C       TRIM STATE CALCULATION
C
        KMAX=NCAP
        DIM=DIMIN+8
        CALL TRIM(MATFILE,NCAP,DIM,DIMIN,DIMINS,KMAX,V,VT,SMT)
        PRINT*
        PRINT*,'CONTINUE WITH FREQUENCY RESPONSE (N/n, DEF=YES)?'
        READ(*,*) CHOICE
        IF (CHOICE.EQ.'N'.OR.CHOICE.EQ.'n') STOP
C
 200    CONTINUE
        PRINT*
        PRINT*,'CONTROL INPUT/TYPE OF INFLOW MODEL TO BE USED:'
        PRINT*
        PRINT*,'      DEFLECTIONS OF ONE FLAP ONLY,'
        PRINT*,'      ALL INFLOW HARMONICS.....MODEL #1'
        PRINT*,'      COLLECTIVE CONTROL INPUT.....MODEL #2'
        PRINT*,'      PROGRESSING CONTROL INPUT.....MODEL #3'
        PRINT*,'      REGRESSING CONTROL INPUT.....MODEL #4'
        PRINT*,'      DIFFERENTIAL CONTROL INPUT.....MODEL #5'
        PRINT*
        PRINT*,'MODEL #?'
        READ(*,*) MODEL
        IF(MODEL.GT.5) THEN
          PRINT*,'INVALID INPUT: MODEL # MAXIMAL 5'
         GOTO 200
        ELSE
        END IF
        IF(MODEL.EQ.1) THEN
          M0=0
          DM=1
           KMAX=(NCAP-M0)/DM
           L=(NCAP-1)/2+1
           DIMINSM=DIMINS
           DIMINM=DIMIN
        ELSE IF(MODEL.EQ.2) THEN
          M0=0
          DM=4
           KMAX=(NCAP-M0)/DM
          KSUM=0
```

```
      DO 7 K=0,KMAX
            KSUM=KSUM+K
7     CONTINUE
        DIMINSM=KMAX*(1+INT(NCAP/2))-2*KSUM
        DIMINM=2*DIMINSM+1+NCAP/2
      ELSE IF(MODEL.EQ.3.OR.MODEL.EQ.4) THEN
        M0=1
        IF(NCAP.LT.M0) THEN
          PRINT*,'M0>NCAP+1: INCREASE MAX. ORDER OF SHAPE FUNCTIONS',
     $          ' NCAP!'
          STOP
        ELSE
        END IF
        DM=2
         KMAX=(NCAP-M0)/DM
        KSUM=0
        DO 8 K=0,KMAX
            KSUM=KSUM+K
8     CONTINUE
        DIMINSM=(1+KMAX)*(1+INT((NCAP-1)/2))-KSUM
        DIMINM=2*DIMINSM
      ELSE
        M0=2
        IF(NCAP.LT.M0) THEN
          PRINT*,'M0>NCAP+1: INCREASE MAX. ORDER OF SHAPE FUNCTIONS',
     $' NCAP!'
          STOP
        ELSE
        END IF
        DM=4
         KMAX=(NCAP-M0)/DM
        KSUM=0
        DO 9 K=0,KMAX
            KSUM=KSUM+K
9     CONTINUE
        DIMINSM=(1+KMAX)*INT(NCAP/2)-2*KSUM
        DIMINM=2*DIMINSM
      END IF
      DIM=DIMINM+8
C
      WRITE(*,600)NRMAX,NRMIN,DNR,SWITCH
      READ(*,*) CHOICE
      IF(CHOICE.EQ.'Y'.OR.CHOICE.EQ.'y') GOTO 201
      PRINT*
      PRINT*,'MINIMUM FREQUENCY, ROTATING SYSTEM [1/REV]?'
      READ(*,*)  NRMIN
      PRINT*,'MAXIMUM FREQUENCY, ROTATING SYSTEM [1/REV]?'
      READ(*,*) NRMAX
      PRINT*,'FREQUENCY RESPONSE CALCULATION STEPWIDTH [1/REV]?'
      READ(*,*)  DNR
      PRINT*,'OUTPUT MODE (PHASE/AMPLITUDE: 1; RE/IM PART:  ELSE)?'
      READ(*,*) SWITCH
```

```
 201   CONTINUE
C
      CALL FRESP(MATFILE,NCAP,MODEL,DIM,DIMINM,DIMINSM,DM,M0,
     $           KMAX,V,NRMIN,NRMAX,DNR,SWITCH)
C
      PRINT*
      PRINT*,'END SESSION.....(DEF)'
      PRINT*,'TASK: MATRIX GENERATION.....(1)'
      PRINT*,'TASK: TRIM CALCULATION.....(2)'
      PRINT*,'TASK: FREQUENCY RESPONSE.....(3)'
      PRINT*
      PRINT*,'CHOICE?'
      READ(*,999) CHOICE
      IF (CHOICE.EQ.'1') THEN
        GOTO 1
      ELSE IF(CHOICE.EQ.'2') THEN
        GOTO 100
      ELSE IF(CHOICE.EQ.'3') THEN
         WRITE(*,700)MATFILE,CT,VINF
        READ(*,*)CHOICE
       IF(CHOICE.EQ.'Y'.OR.CHOICE.EQ.'y') GOTO 301
 300    CONTINUE
        PRINT*,'MATRIX FILE TO BE USED (A8)?'
        READ(*,*) MATFILE
        IF(MATFILE(7:).NE.'MA')THEN
             PRINT*,'INPUT ERROR: FILE NAME MUST END WITH "MA"!'
             GOTO 300
       ELSE
       END IF
       PRINT*,'THRUST COEFFICIENT CT?'
       READ(*,*) CT
       PRINT*,'RATE OF CLIMB?'
       READ(*,*) VINF
 301    CONTINUE
         NCAP=10*( ICHAR(MATFILE(5:))-48 )+ICHAR(MATFILE(6:))-48
       IF(VINF.GE.0)THEN
            SMT=(-VINF/2.+SQRT(VINF/2.*VINF/2.+CT/2.))/SQRT(3.)
       ELSE
            SMT=(-VINF/2.-SQRT(VINF/2.*VINF/2.+CT/2.))/SQRT(3.)
       END IF
        VT=VINF+2.*SQRT(3.)*SMT
       GOTO 200
      ELSE
      END IF
      PRINT*,'BYE!'
 500  FORMAT(' USE DEFAULT CT=',F7.4,' AND VINF=',F7.4,' (Y/y)?')
 600  FORMAT(' USE DEFAULT NRMAX=',F7.3,', NRMIN=',
     $       F7.3,', DNR=',F7.3,' AND SWITCH=',I2,' (Y/y)?')
 700  FORMAT(' USE SETTINGS MATFILE=',A8,', CT=',F7.4,', AND VINF=',
     $       F7.4,'  (Y/y)?')
 999  FORMAT(A1)
      END
```

Matrix Assembly:

File 'matgen.f'

```
      SUBROUTINE MATGEN(DATASET, NCAP, DIMIN, DIMINC)
C*****************************************************************
C
C     GENERATION OF FILE MATFILE USING ROTOR DATA FROM
C     FILE DATASET
C
C     CALCULATION OF SUBMATRICES TO BE USED FOR ASSEMBLING
C     SYSTEM DYNAMICS MATRICES FOR ALL ALLOWED COMBINATIONS OF
C     HARMONIC M = 0,..., NCAP AND POLYNOMIAL NUMBER N
C
C     VERSION 1.2, NOVEMBER 29, 1990:
C           UTILIZE SYMMETRY OF BMNJ AND DKII (MEMORY REDUCTION)
C
C*****************************************************************
      INTEGER NCAP, DIMIN, DIMINC
      INTEGER I, IMAX, J, JMAX, MI, MJ, NI, NJ, L, K, S
      REAL C, CF, CETAF,R0, R1, R2, R3, E, EA, EF, TT1, TT2, T0, TBAR,BBAR
      REAL A, KBF, KF, CLD, CL0S, CMD, CM0B, CM0BF, GAMMA
      REAL IBT, IBD, ITT, ITD, IDD, SB, ST, SD, KBE, KC, KT, CETA, ETA
      REAL KBB, KBT, KBD, KTT, KTD
      REAL I11R2R3, I12R2R3, I13R2R3
      REAL I21R2R3, I22R2R3
      REAL I31R2R3, I32R2R3, I41R2R3, I42R2R3
      REAL I31R0R2, I32R0R2, I31R31, I32R31
      REAL I31R11, I41R11, I32R11, I42R11
      REAL I41R0R1, I42R0R1
      REAL INT12, INT13, INT21, INT22, INT23, INT31, INT32, INT42
      REAL J1R2R3, J2R01, J2R2R3
      REAL J3R01, J3R0R2, J3R2R3, J3R11, J3R31
      REAL J4R0R1, J4R2R3, J4R11
      REAL J5R01, J5R2R3, J6R01, J6R2R3, J7R01, J7R2R3
      REAL PI, SUM, EXPO, KMNMIS, KMNMINI, KMNMIJ
      REAL DUM(0:20)
      REAL DUMM(0:20,0:20)
      CHARACTER*8 MATFILE
      CHARACTER*4 DATASET
C
      EXTERNAL FAC2
C
C
C
      PRINT*,'MODULE MATGEN'
      PRINT*
C
      OPEN(UNIT=7,FILE = DATASET)
      READ(7,*) DATASET
```

```
      READ(7,500) C
      READ(7,500) CF
      READ(7,500) CETAF
      READ(7,500) R0
      READ(7,500) R1
      READ(7,500) R2
      READ(7,500) R3
      READ(7,500) E
      READ(7,500) EA
      READ(7,500) EF
      READ(7,500) TT1
      READ(7,500) TT2
      READ(7,500) T0
      READ(7,500) TBAR
      READ(7,500) BBAR
      READ(7,500) A
      READ(7,500) KBF
      READ(7,500) KF
      READ(7,500) CLD
      READ(7,500) CL0B
      READ(7,500) CMD
      READ(7,500) CM0B
      READ(7,500) CM0BF
      READ(7,500) GAMMA
      READ(7,500) IBT
      READ(7,500) IBD
      READ(7,500) ITT
      READ(7,500) ITD
      READ(7,500) IDD
      READ(7,500) SB
      READ(7,500) ST
      READ(7,500) SD
      READ(7,500) KBE
      READ(7,500) KC
      READ(7,500) KT
      READ(7,500) CETA
      READ(7,500) ETA
      CLOSE(7)
C
      PRINT*,'     MATGEN: READ GEOMETRIC DATA FROM FILE ',DATASET
C
      MATFILE=DATASET
      MATFILE(5:)=CHAR(NCAP/10+48)
      MATFILE(6:)=CHAR(NCAP-(NCAP/10)*10+48)
      MATFILE(7:)='MA'
C
C     INTEGRAL EXPRESSIONS
C
      CALL ANALY(E, R0, R1, R2, R3, KBF,
     $           I11R2R3, I12R2R3, I13R2R3, I21R2R3, I22R2R3,
     $           I31R2R3, I32R2R3, I41R2R3, I42R2R3,
     $           I31R11, I32R11, I41R11, I42R11,
```

```
     $            I31R0R2, I32R0R2, I31R31, I32R31,
     $            I41R0R1, I42R0R1,
     $            INT12, INT13, INT21, INT22, INT23, INT31, INT32, INT42)
          PI=4.*ATAN(1.)
C
C
          PRINT*,'      MATGEN: CALCULATED ANALYTIC INTEGRALS'
C
          OPEN(UNIT=8, FILE=MATFILE)
          WRITE(8,*) MATFILE
C
C         MBB
C
          WRITE(8,800)
          DUM(1)=1.-ETA*IBD
     $          +GAMMA*C/8.*INT13+CF/8.*KF*I13R2R3
          DUM(2)=-IBT-CETA*IBD
     $          -GAMMA*CF/8.*EF*C*KF*I12R2R3
          WRITE(8,510)  DUM(1),DUM(2)
          DUM(1)=-IBT+ETA*ITD
     $          -GAMMA*(C*C/32.*(EA+0.25)*INT12
     $                +CF*CF/32.*KF*(EF/CETAF+0.25)*I12R2R3)
          DUM(2)=ITT+CETA*ITD
     $          +GAMMA*CF*CF/32.*EF*C*KF*(EF/CETAF+0.25)*I11R2R3
          WRITE(8,510)  DUM(1),DUM(2)
C
          PRINT*,'      MATGEN: CALCULATED MBB'
C
C
C         MIBT
C
          WRITE(8,810)
          DO 10 MI=0,NCAP
            IMAX = (NCAP-MI)/2
            WRITE(8,700) MI
           DO 5 I=0,IMAX
                NI=MI+2*I+1
                CALL  NUM(J5R01,5,R0,1.,NI,MI,L,K,E)
                CALL  NUM(J5R2R3,5,R2,R3,NI,MI,L,K,E)
                CALL  NUM(J1R2R3,1,R2,R3,NI,MI,L,K,E)
                DUM(1)=A*C/(16.*PI)
     $                *(C*(J5R01+(KBF-1.)*J5R2R3)
     $                  +CF*KF*J5R2R3)
                DUM(2)=-A*C/(16.*PI)*CF*EF*C*KF*J1R2R3
                WRITE(8,520)  NI,DUM(1),DUM(2)
 5         CONTINUE
 10       CONTINUE
C
          PRINT*,'      MATGEN: CALCULATED MIBT'
C
C         CBB
C
```

```
      WRITE(8,820)
      DUM(1)=GAMMA/2.*INT23  -  ETA*GAMMA*CF/8.*KF*I22R2R3
      DUM(2)=-GAMMA/2.*(C*INT22+CF/4.*KF*I22R2R3)
     $        -CETA*GAMMA*CF/8.*KF*I22R2R3
      WRITE(8,510)  DUM(1),DUM(2)
      DUM(1)=-GAMMA*EA*C/2.*INT22
     $        +ETA*GAMMA*CF*CF/32.*KF*(EF/CETAF+0.25)*I21R2R3
      DUM(2)=GAMMA
     $        *(C*C/8.*(3.*EA+0.25)*INT21
     $        +(C*C*KBF*(EA+0.25)+(C*C*KBF+CF*CF*KF)*(EF/CETAF+0.25))/32.
     $          *I21R2R3)
     $        +CETA*GAMMA*CF*CF/32.*KF*(EF/CETAF+0.25)*I21R2R3
      WRITE(8,510)  DUM(1),DUM(2)
C
      PRINT*,'      MATGEN: CALCULATED CBB'
C
C
C     CIBT
C
      WRITE(8,830)
      DO 20 MI=0,NCAP
        IMAX = (NCAP-MI)/2
        WRITE(8,700) MI
       DO 15 I=0,IMAX
          NI=MI+2*I+1
          CALL  NUM(J6R01,6,R0,1.,NI,MI,L,K,E)
          CALL  NUM(J6R2R3,6,R2,R3,NI,MI,L,K,E)
          CALL  NUM(J2R01,2,R0,1.,NI,MI,L,K,E)
          CALL  NUM(J2R2R3,2,R2,R3,NI,MI,L,K,E)
          DUM(1)=A*C/(4.*PI)*
     $          (J6R01+(KBF-1.)*J6R2R3)
     $          -ETA*A*C/(16.*PI)*CF*KF*J2R2R3
          DUM(2)=-A*C/(4.*PI)*
     $          (C*(J2R01+(KBF-1.)*J2R2R3)
     $           +CF*KF*J2R2R3/4.)
     $          -CETA*A*C/(16.*PI)*CF*KF*J2R2R3
          WRITE(8,520)  NI,DUM(1),DUM(2)
 15     CONTINUE
 20   CONTINUE
C
      PRINT*,'      MATGEN: CALCULATED CIBT'
C
C
C     KBB
C
      WRITE(8,840)
      KBB=1.+SB*E+KBE
      KBT=IBT+ST*E+KC
      KBD=IBD+SD*E
      KTT=ITT+2.*SD*EF*C+KT
      KTD=IDD+SD*EF*C
C
```

```
      DUM(1)= KBB-ETA*KBD
     $        -GAMMA*3./8.*C*INT22
     $        -ETA*GAMMA*CLD/(2.*A)*I32R2R3
      DUM(2)= -KBT-CETA*KBD
     $        -GAMMA/2.*INT32
     $        -CETA*GAMMA*CLD/(2.*A)*I32R2R3
      WRITE(8,510)  DUM(1),DUM(2)
      DUM(1)= -KBT+ETA*KTD
     $        +GAMMA*C*C/8.*(3.*EA+0.25)*INT21
     $        -ETA*GAMMA*CMD/(2.*A)*C*I31R2R3
      DUM(2)= KTT+CETA*KTD
     $        +GAMMA/2.*EA*C*INT31
     $        -CETA*GAMMA*CMD/(2.*A)*C*I31R2R3
      WRITE(8,510)  DUM(1),DUM(2)
C
      PRINT*,'      MATGEN: CALCULATED KBB'
C
C
C     KBIT TRANSPOSED
C
      WRITE(8,850)
      DO 30 MI=0,NCAP
        IMAX = (NCAP-MI)/2
        WRITE(8,700) MI
       DO 25 I=0,IMAX
            NI=MI+2*I+1
            CALL  NUM(J6R01,6,R0,1.,NI,MI,L,K,E)
            CALL  NUM(J6R2R3,6,R2,R3,NI,MI,L,K,E)
            CALL  NUM(J2R01,2,R0,1.,NI,MI,L,K,E)
            CALL  NUM(J2R2R3,2,R2,R3,NI,MI,L,K,E)
            DUM(1)=  GAMMA/2.*
     $             (J6R01+(KBF-1.)*J6R2R3)
            DUM(2)=-GAMMA/2.*C*EA*
     $             (J2R01+(KBF-1.)*J2R2R3)
            WRITE(8,520)  NI,DUM(1),DUM(2)
 25     CONTINUE
 30   CONTINUE
C
      PRINT*,'      MATGEN: CALCULATED KBIT'
C
C
C     KIBT
C
      WRITE(8,860)
      DO 40 MI=0,NCAP
        IMAX = (NCAP-MI)/2
        WRITE(8,700) MI
       DO 35 I=0,IMAX
            NI=MI+2*I+1
            CALL  NUM(J2R01,2,R0,1.,NI,MI,L,K,E)
            CALL  NUM(J2R2R3,2,R2,R3,NI,MI,L,K,E)
            CALL  NUM(J3R01,3,R0,1.,NI,MI,L,K,E)
```

```
            CALL  NUM(J3R2R3,3,R2,R3,NI,MI,L,K,E)
            DUM(1)=-A*C/(16.*PI)*3.*C*
     $             (J2R01+(KBF-1.)*J2R2R3)
     $             -ETA*A*C/(4.*PI)*CLD/A*J3R2R3
            DUM(2)=-A*C/(4.*PI)*
     $             (J3R01+(KBF-1.)*J3R2R3)
     $             -CETA*A*C/(4.*PI)*CLD/A*J3R2R3
            WRITE(8,520)  NI,DUM(1),DUM(2)
 35       CONTINUE
 40     CONTINUE
C
       PRINT*,'      MATGEN: CALCULATED KIBT'
C
C
C      KNM
C
       WRITE(8,870)
       DO 50 MI=0,NCAP
          IMAX = (NCAP-MI)/2
          WRITE(8,700) MI
         DO 45 I=0,IMAX
            NI=MI+2*I+1
            DUM(1)=2./PI*(FAC2(NI+MI-1)*FAC2(NI-MI-1))
     $                  /(FAC2(NI+MI)*FAC2(NI-MI))
            WRITE(8,520)  NI,DUM(1)
 45       CONTINUE
 50     CONTINUE
C
       PRINT*,'      MATGEN: CALCULATED KNM'
C
C
C      BMNJ
C
       WRITE(8,880)
       DO 70 MI=0,NCAP
          IMAX = (NCAP-MI)/2
          WRITE(8,700) MI
         DO 65 I=0,IMAX
            NI=MI+2*I+1
            KMNMINI=2./PI*(FAC2(NI+MI-1)*FAC2(NI-MI-1))
     $                  /(FAC2(NI+MI)*FAC2(NI-MI))
            DO 60 K=I, IMAX
              J=MI+2*K+1
                KMNMIJ=2./PI*(FAC2(J+MI-1)*FAC2(J-MI-1))
     $                  /(FAC2(J+MI)*FAC2(J-MI))
              SUM=0.
               EXPO=(J+NI-2.*MI-2.)/2.
              DO 55 S=MI,NI-1,2
                KMNMIS=2./PI*(FAC2(S+MI-1)*FAC2(S-MI-1))
     $                  /(FAC2(S+MI)*FAC2(S-MI))
                SUM=SUM+KMNMIS*(2.*S+1.)/((J-S)*(J+S+1.))
 55           CONTINUE
```

```
                DUMM(I,K)=PI/2.*SQRT(KMNMINI/KMNMIJ)
     $                    *(-1.)**EXPO
     $                    *SQRT((2.*J+1.)*(2.*NI+1.))  *SUM
                DUMM(K,I)=DUMM(I,K)
 60           CONTINUE
              WRITE(8,530)  NI,(DUMM(I,K),K=0,IMAX)
 65       CONTINUE
 70     CONTINUE
C
       PRINT*,'      MATGEN: CALCULATED BMNJ'
C
C
C      DKII
C
       WRITE(8,890)
       DO 90 MI=0,NCAP
         IMAX = (NCAP-MI)/2
         DO 85 MJ=MI,NCAP
              JMAX=(NCAP-MJ)/2
              WRITE(8,710)  MI,MJ
              DO 80 I=0,IMAX
                NI=MI+2*I+1
                DO 75 J=0,JMAX
                     NJ=MJ+2*J+1
                     CALL  NUM(J7R01,7,R0,1.,NI,MI,NJ,MJ,E)
                     CALL  NUM(J7R2R3,7,R2,R3,NI,MI,NJ,MJ,E)
                     DUM(J+1)=   A*C/(4.*PI)*
     $                    (J7R01+(KBF-1.)*J7R2R3)
 75              CONTINUE
                 WRITE(8,530)  NI,(DUM(K+1),K=0,JMAX)
 80           CONTINUE
 85       CONTINUE
 90     CONTINUE
C
       PRINT*,'      MATGEN: CALCULATED DKII'
C
C
C      RB
C
       WRITE(8,900)
       DUM(1)=IBD
       DUM(2)=GAMMA/8.*CF*KF*I22R2R3
       DUM(3)=KBD+GAMMA/2.*CLD/A*I32R2R3
       WRITE(8,510)  (DUM(I),I=1,3)
       DUM(1)=-ITD
       DUM(2)=-GAMMA/32.*KF*CF*CF*(EF/CETAF+0.25)*I21R2R3
       DUM(3)=-KTD+GAMMA/2.*C*CMD/A*I31R2R3
       WRITE(8,510)  (DUM(I),I=1,3)
C
       PRINT*,'      MATGEN: CALCULATED RB'
C
C
```

```
C       RIT
C
        WRITE(8,910)
        DO 100 MI=0,NCAP
          IMAX = (NCAP-MI)/2
          WRITE(8,700) MI
         DO 95 I=0,IMAX
              NI=MI+2*I+1
              CALL  NUM(J2R2R3,2,R2,R3,NI,MI,L,K,E)
              CALL  NUM(J3R2R3,3,R2,R3,NI,MI,L,K,E)
              DUM(1)=0.
              DUM(2)=A*C/(16.*PI)*CF*KF*J2R2R3
              DUM(3)=A*C/(4.*PI)*CLD/A*J3R2R3
              WRITE(8,520)  NI,DUM(1),DUM(2),DUM(3)
 95       CONTINUE
 100    CONTINUE
C
        PRINT*,'       MATGEN: CALCULATED RIT
C
C
C       COB
C
        WRITE(8,920)
        DUM(1)=GAMMA/2.*(
     $                  (CL0B/A+T0)*(I32R0R2+I32R31)
     $                      +KBF*T0*I32R2R3
     $                 -TT1*I42R0R1
     $                 -TT2*(I42R11+(KBF-1.)*I42R2R3)
     $                 -(TT1-TT2)*R1*(I32R11+(KBF-1.)*I32R2R3)
     $                 )
     $          +KBE*BBAR-KC*TBAR
        WRITE(8,510)  DUM(1)
        DUM(1)=GAMMA/2.*C
     $                 *((CM0B/A-T0*EA)*(I31R0R2+I31R31)
     $                       -KBF*T0*EA*I31R2R3
     $                   +EA*TT1*I41R0R1
     $                    +EA*TT2*(I41R11+(KBF-1.)*I41R2R3)
     $                    +EA*(TT1-TT2)*R1*(I31R11+(KBF-1.)*I31R2R3)  )
     $           +KT*TBAR-KC*BBAR
        WRITE(8,510)  DUM(1)
C
        PRINT*,'       MATGEN: CALCULATED COB'
C
C
C       COIT
C
        WRITE(8,930)
        DO 110 MI=0,NCAP
          IMAX = (NCAP-MI)/2
          WRITE(8,700) MI
         DO 105 I=0,IMAX
              NI=MI+2*I+1
```

```
            CALL  NUM(J3R11,3,R1,1.,NI,MI,L,K,E)
            CALL  NUM(J3R2R3,3,R2,R3,NI,MI,L,K,E)
            CALL  NUM(J3R0R2,3,R0,R2,NI,MI,L,K,E)
            CALL  NUM(J3R31,3,R3,1.,NI,MI,L,K,E)
            CALL  NUM(J4R0R1,4,R0,R1,NI,MI,L,K,E)
            CALL  NUM(J4R11,4,R1,1.,NI,MI,L,K,E)
            CALL  NUM(J4R2R3,4,R2,R3,NI,MI,L,K,E)
            DUM(1)=A*C/(4.*PI)
     $        *((CL0B/A+T0)*(J3R0R2+J3R31)
     $             +KBF*T0*J3R2R3
     $          -TT1*J4R0R1
     $           -TT2*(J4R11+(KBF-1.)*J4R2R3)
     $          -(TT1-TT2)*R1
     $            *(J3R11+(KBF-1.)*J3R2R3))
          WRITE(8,520)  NI,DUM(1)
 105    CONTINUE
 110  CONTINUE
C
      PRINT*,'      MATGEN: CALCULATED COIT'
C
C
      CLOSE(8)
C
C
C
      PRINT*
      PRINT*,'      MATGEN: FINISHED MATRIX GENERATION;'
      PRINT*,'                        RESULTS STORED IN FILE  ',MATFILE
      PRINT*
C
 500  FORMAT(18X,E12.6)
 510  FORMAT(3E14.6)
 520  FORMAT(I3,3E14.6)
 530  FORMAT(I3,20E14.6)
C
 700  FORMAT(/,'MI=',I3)
 710  FORMAT(/,'MI=',I3,/,'MJ=',I3)
C
 800  FORMAT(//,'MBB')
 810  FORMAT(//,'MIBT')
 820  FORMAT(//,'CBB')
 830  FORMAT(//,'CIBT')
 840  FORMAT(//,'KBB')
 850  FORMAT(//,'KBIT  TRANSPOSED')
 860  FORMAT(//,'KIBT')
 870  FORMAT(//,'KNM')
 880  FORMAT(//,'BMNJ')
 890  FORMAT(//,'DKII')
 900  FORMAT(//,'RB')
 910  FORMAT(//,'RIT')
 920  FORMAT(//,'COB')
 930  FORMAT(//,'COIT')
```

```
      RETURN
      END


      REAL FUNCTION FAC2(N)
C
      INTEGER N, IDIM
      REAL N2FAC
      REAL RDIM
      IF(N.LE.0) THEN
        IF(N.EQ.0.OR.N.EQ.-1) THEN
            FAC2 = 1.
            RETURN
        ELSE IF (N.EQ.-3) THEN
            FAC2 = -1.
            RETURN
        ELSE
            PRINT*,'  ERROR IN FAC2: N=',N,' CANNOT BE COMPUTED!'
            STOP
        END IF
      ELSE
      END IF
      IDIM = N/2
      RDIM = N/2.
      IF (RDIM-IDIM.EQ.0) THEN
        IDIM = IDIM -1
      ELSE
      END IF
      N2FAC=N
      DO 10 I = 1, IDIM
        N2FAC=N2FAC*(N-I*2)
 10   CONTINUE
      FAC2=N2FAC
      RETURN
      END

      SUBROUTINE ANALY(E,R0,R1,R2,R3,KBF,
     $                 I11R2R3, I12R2R3, I13R2R3, I21R2R3, I22R2R3,
     $                 I31R2R3, I32R2R3, I41R2R3, I42R2R3,
     $                 I31R11, I32R11, I41R11, I42R11,
     $                 I31R0R2, I32R0R2, I31R31, I32R31,
     $                 I41R0R1, I42R0R1,
     $                 INT12, INT13, INT21, INT22, INT23, INT31, INT32, INT42)
C*****************************************************************
C
C     ANALYTICAL COMPUTATION OF INTEGRALS NEEDED IN MATRICES
C
C*****************************************************************
      REAL E, R0, R1, R2, R3, KBF
      REAL I11R2R3, I12R2R3, I13R2R3, I21R2R3, I22R2R3
      REAL I31R2R3, I32R2R3, I41R2R3, I42R2R3
      REAL I31R11, I32R11, I41R11, I42R11
```

```
      REAL I31R0R2, I32R0R2, I31R31, I32R31
      REAL I41R0R1, I42R0R1
      REAL INT12, INT13, INT21, INT22, INT23, INT31, INT32, INT42
      REAL B, T
C
      ZI11(B,T)=T-B
      ZI21(B,T)=0.5*(T*T-B*B)
      ZI31(B,T)=1./3.*(T*T*T-B*B*B)
      ZI41(B,T)=0.25*(T*T*T*T-B*B*B*B)
      ZI12(B,T)=ZI21(B,T)-E*ZI11(B,T)
      ZI22(B,T)=ZI31(B,T)-E*ZI21(B,T)
      ZI32(B,T)=ZI41(B,T)-E*ZI31(B,T)
      ZI42(B,T)=0.2*(T*T*T*T*T-B*B*B*B*B)-E*ZI41(B,T)
      ZI13(B,T)=ZI22(B,T)-E*ZI12(B,T)
      ZI23(B,T)=ZI32(B,T)-E*ZI22(B,T)
C
      I11R2R3=ZI11(R1,R3)
      I12R2R3=ZI12(R2,R3)
      I13R2R3=ZI13(R2,R3)
      I21R2R3=ZI21(R2,R3)
      I22R2R3=ZI22(R2,R3)
      I31R2R3=ZI31(R2,R3)
      I32R2R3=ZI32(R2,R3)
      I41R2R3=ZI41(R2,R3)
      I42R2R3=ZI42(R2,R3)
C
      I31R11  =ZI31(R1,1.)
      I32R11  =ZI32(R1,1.)
      I41R11  =ZI41(R1,1.)
      I42R11  =ZI42(R1,1.)
C
      I31R31  =ZI31(R3,1.)
      I32R31  =ZI32(R3,1.)
C
      I41R0R1=ZI41(R0,R1)
      I42R0R1=ZI42(R0,R1)
      I31R0R2=ZI31(R0,R2)
      I32R0R2=ZI32(R0,R2)
C
      INT12=ZI12(R0,1.)+(KBF-1.)*I12R2R3
      INT13=ZI13(R0,1.)+(KBF-1.)*I13R2R3
      INT21=ZI21(R0,1.)+(KBF-1.)*I21R2R3
      INT22=ZI22(R0,1.)+(KBF-1.)*I22R2R3
      INT23=ZI23(R0,1.)+(KBF-1.)*I23R2R3
      INT31=ZI31(R0,1.)+(KBF-1.)*I31R2R3
      INT32=ZI32(R0,1.)+(KBF-1.)*I32R2R3
      INT42=ZI42(R0,1.)+(KBF-1.)*I42R2R3
      RETURN
      END
```

```
      SUBROUTINE NUM(INT,I,A,B,N,M,L,K,E)
C*****************************************************************
C
C     RETURNS AS SS THE INTEGRAL OF THE FUNCTION FUNC BETWEEN
C     A AND B, BY TEN-POINT GAUSS-LEGENDRE INTEGRATION;
C     THE FUNCTION IS EVALUATED EXACTLY TEN TIMES AT INTERIOR
C     POINTS IN THE RANGE OF INTEGRATION
C
C     SOURCE: 'NUMERICAL RECIPES'(QGAUS), MODIFIED
C
C     VERSION OCTOBER 17,1990
C
C     MARTIN STETTNER
C
C*****************************************************************
      DIMENSION X(5),W(5)
      REAL E,INT
      INTEGER I,N,M,L,K
      EXTERNAL FUNC1, FUNC2, FUNC3, FUNC4, FUNC5, FUNC6, FUNCK
C     abscissas and weights
      DATA X/.1488743389,.4333953941,.6794095682,
     $              .8650633666,.9739065285/
      DATA W/.2955242247,.2692667193,.2190863625,
     $              .1494513491,.0666713443/
C
      XM=0.5*(B+A)
      XR=0.5*(B-A)
      SS=0.
      DO 10 J=1,5
            DX=XR*X(J)
            IF (I.EQ.1) THEN
                  SS=SS+W(J)*(FUNC1(XM+DX,N,M)
     $                        +FUNC1(XM-DX,N,M))
            ELSE IF (I.EQ.2) THEN
                  SS=SS+W(J)*(FUNC2(XM+DX,N,M)
     $                        +FUNC2(XM-DX,N,M))
            ELSE IF (I.EQ.3) THEN
                  SS=SS+W(J)*(FUNC3(XM+DX,N,M)
     $                        +FUNC3(XM-DX,N,M))
            ELSE IF (I.EQ.4) THEN
                  SS=SS+W(J)*(FUNC4(XM+DX,N,M)
     $                        +FUNC4(XM-DX,N,M))
            ELSE IF (I.EQ.5) THEN
                  SS=SS+W(J)*(FUNC5(XM+DX,N,M,E)
     $                        +FUNC5(XM-DX,N,M,E))
            ELSE IF (I.EQ.6) THEN
                  SS=SS+W(J)*(FUNC6(XM+DX,N,M,E)
     $                        +FUNC6(XM-DX,N,M,E))
            ELSE
```

```
                SS=SS+W(J)*(FUNCK(XM+DX,N,M,L,K)
     $                 +FUNCK(XM-DX,N,M,L,K))
           END IF
10    CONTINUE
      INT=XR*SS
      RETURN
      END

      REAL FUNCTION FUNC1(R,N,M)
      REAL R, NU
      INTEGER N, M
      EXTERNAL PNM
      NU = SQRT(1.0-R*R)
      FUNC1 = PNM(N,M,NU,2)
      RETURN
      END

      REAL FUNCTION FUNC2(R,N,M)
      REAL R, NU
      INTEGER N, M
      EXTERNAL PNM
      NU = SQRT(1.0-R*R)
      FUNC2 = PNM(N,M,NU,2)*R
      RETURN
      END

      REAL FUNCTION FUNC3(R,N,M)
      REAL R, NU
      INTEGER N, M
      EXTERNAL PNM
      NU = SQRT(1.0-R*R)
      FUNC3 = PNM(N,M,NU,2) * R*R
      RETURN
      END

      REAL FUNCTION FUNC4(R,N,M)
      REAL R, NU
      INTEGER N, M
      EXTERNAL PNM
      NU = SQRT(1.0-R*R)
      FUNC4 = PNM(N,M,NU,2) *R*R*R
      RETURN
      END

      REAL FUNCTION FUNC5(R,N,M,E)
      REAL R, NU, E
      INTEGER N, M
      EXTERNAL PNM
      NU = SQRT(1.0-R*R)
      FUNC5 = PNM(N,M,NU,2) *(R-E)
      RETURN
      END
```

```
      REAL FUNCTION FUNC6(R,N,M,E)
      REAL R, NU, E
      INTEGER N, M
      EXTERNAL PNM
      NU = SQRT(1.0-R*R)
      FUNC6 = PNM(N,M,NU,2) *R*(R-E)
      RETURN
      END

      REAL FUNCTION FUNCK(R,N,M,L,K)
      REAL R, NU
      INTEGER N, M, L, K
      EXTERNAL PNM
      NU = SQRT(1.0-R*R)
      FUNCK = PNM(N,M,NU,2)*PNM(L,K,NU,2)*R
      RETURN
      END

      REAL FUNCTION PNM (N,M,X,L)
C.....
C HE/PETERS, MARCH 1988 (MOD OCTOBER 1990: DISCARD ERROR MESSAGE)
C
C COMPUTES PNM(X) FOR ANY N+M COMBINATION,
C FOR PNM(X)/X, N+M SHOULD BE ODD ONLY
C PNM(X) IS THE NORMALIZED ASSOCIATED LEGENDRE FUNCTION OF THE
C  FIRST KIND (NALF).
C N:   THE SUBSCRIPT OF THE NALF (HARMONIC NUMBER)
C M:   THE SUPERSCRIPT OF THE NALF (POLYNOMIAL NUMBER)
C X:   THE INDEPENDENT VARIABLE OF THE LEGENDRE FUNCTIONS.
C      IN PETER' THEORY IT IS THE ELLIPSOIDAL COORDINATE. AT THE
C      ROTOR DISC X=SQRT(1.0-R*R) WHERE R IS THE NONDIMENSIONAL
C      RADIAL COORDINATE.
C L:   CONTROL SWITCH. NOTE THAT FOR N+M EVEN, USE L=1 ONLY!!
C      L=1    EVALUATE PNM(X)
C      L=2    EVALUATE PNM(X)/X
C.....
      REAL  PA(0:20)
      NPM=N-M
      IF (NPM.LT.0) THEN
        PNM=0.0
        RETURN
      END IF
      MAN = M+N
      MATEST = (MAN/2)*2-MAN
      IF (MATEST.EQ.0.AND.L.EQ.2) THEN
        PRINT*,' ERROR IN PNM: L DOES NO MATCH N+M ODD/EVEN!'
        STOP
      END IF
      PA(1) = PM1M(M,X,2)
      PA(2) = PM2M(M,X)
      PA(0) = PMM(M,X)
```

```
      NCT = N-M+1
      IF (NCT.LT.4) THEN
        GOTO (1,2,3), NCT
1       PNM=PA(0)
        RETURN
2       GOTO (28,29),L
28      PNM = PA(1)*X
        RETURN
29      PNM = PA(1)
        RETURN
3       PNM = PA(2)
        RETURN
      END IF
      JC = 2
      DO 30 I = M+3,N
         DD = (I+M)*(I-M)
          CNT1=(2*I+1.0)*(2*I-1.0)/DD
        CNT1=SQRT(CNT1)
            CNT2=(2*I+1.)*(I-1.-M)*(I-1.+M)/((2.*I-3.)*DD)
        CNT2=SQRT(CNT2)
        JC=JC+1
          IC=(JC-1)/2-JC/2
        IF(IC.EQ.0) THEN
              PA(JC)=CNT1*PA(JC-1)-CNT2*PA(JC-2)
        ELSE
              PA(JC)=CNT1*X*X*PA(JC-1)-CNT2*PA(JC-2)
        END IF
30    CONTINUE
      NM2=(N+M)/2
      NTE=2*NM2-(N+M)
      IF(NTE.EQ.0) THEN
        PNM=PA(JC)
        RETURN
      ELSE
        GOTO(61,62),L
61      PNM=PA(JC)*X
        RETURN
62      PNM=PA(JC)
      END IF
      RETURN
      END

      REAL FUNCTION PM1M(M,X,L)
      A=SQRT(3.)
      IF(M.EQ.0) GOTO 100
      A=1.
      DO 11 I=1,M
         CC=(2.*I+1.)/(2.*I)
        AA=SQRT(CC)*SQRT(1.-X*X)
11    A=A*AA
      A=A*SQRT(2.*M+3.)
100   GOTO(10,20),L
```

```
10    CONTINUE
      PM1M=A*X
      RETURN
20    PM1M=A
      RETURN
      END

      REAL FUNCTION PM3M(M,X,L)
      IF(M.EQ.0) THEN
        Q=0.5*7.*6.*3.
       Q=SQRT(Q)
        C=Q*(5./6.*X*X-0.5)
       GOTO 100
      ELSE
      ENDIF
      A=1.
      DO 11 I=1,M
         CC=(2.*I+1.)/(2.*I)
       AA=SQRT(CC)*SQRT(1.-X*X)
11    A=A*AA
      EE=(2.*M+7.)*6.*(2.*M+3.)/(2.*M+2.)
      EE=SQRT(EE)
      B=A*EE
      BB=X*X*(2.*M+5.)/6.-0.5
      C=B*BB
100   GOTO(10,20),L
10    PM3M=C*X
      RETURN
20    PM3M=C
      RETURN
      END

      REAL FUNCTION PM2M(M,X)
      A=2.*M
      C1=3.*(A+3.)/((A+7.)*(A+5.))
      C1=SQRT(C1)
      C2=2.*(A+2.)/((A+3.)*(A+5.))
      C2=SQRT(C2)
      PM2M=C1*PM3M(M,X,2)+C2*PM1M(M,X,2)
      RETURN
      END

      REAL FUNCTION PMM(M,X)
      A1=(2.*M+5.)/(4.*(M+1.))
      A1=SQRT(A1)
      B1=(2.*M+5.)*(2.*M+3.)/(4.*(M+1.))
      B1=SQRT(B1)
      C1=B1/A1
      C2=1./A1
      PMM=C1*X*PM1M(M,X,1)-C2*PM2M(M,X)
      RETURN
      END
```

Sample Input:

Data File for Kaman 101 Main Rotor

```
K101
C          0.836000E-01
CF         0.322000E-01
CETAF      0.385200E+00
R0         0.210700E+00
R1         0.525000E+00
R2         0.655400E+00
R3         0.824600E+00
E          0.313000E-01
EA         0.211000E-01
EF         0.845800E+00
TT1        0.392400E+00
TT2        0.113100E+00
T0         0.231457E+00
TBAR       0.471000E+00
BBAR       0.000000E+00
A          0.683900E+01
KBF        0.957100E+00
KF         0.408600E+00
CLD        0.337200E+01
CL0B       0.136700E+00
CMD        -.612000E+00
CM0B       -.400000E-02
CM0BF      0.200000E-01
GAMMA      0.662700E+01
IBT        0.246900E-02
IBD        0.138400E-03
ITT        0.102300E-02
ITD        0.636800E-05
IDD        0.636800E-05
SB         0.145320E+01
ST         0.907790E-02
SD         0.156990E-03
KBE        0.000000E+00
KC         0.000000E+00
KT         0.570730E-03
CETA       0.410000E+00
ETA        -.430000E+00
```

Sample Output:

Matrix File for four Radial Expansion Functions

(N = 4)

```
 K10104MA


MBB
  0.102080E+01 -0.261815E-02
 -0.267324E-02  0.103016E-02


MIBT

MI= 0
  1  0.768865E-03 -0.310112E-05
  3 -0.377753E-03  0.177622E-05
  5  0.114369E-03  0.253352E-05

MI= 1
  2  0.847559E-03 -0.362844E-05
  4 -0.132159E-03 -0.259302E-06

MI= 2
  3  0.850183E-03 -0.356746E-05
  5 -0.287511E-05 -0.190206E-05

MI= 3
  4  0.834638E-03 -0.326128E-05

MI= 4
  5  0.814818E-03 -0.286650E-05


CBB
  0.760216E+00 -0.876403E-01
 -0.183186E-02  0.136603E-02


CIBT

MI= 0
  1  0.245564E-01 -0.315816E-02
  3 -0.196874E-01  0.147504E-02
  5  0.843585E-02 -0.542171E-03

MI= 1
```

```
  2  0.294616E-01 -0.345191E-02
  4 -0.126819E-01  0.500480E-03

MI=  2
  3  0.313286E-01 -0.345021E-02
  5 -0.800813E-02  0.153814E-04

MI=  3
  4  0.320694E-01 -0.338501E-02

MI=  4
  5  0.323066E-01 -0.330777E-02


KBB
  0.102759E+01 -0.830423E+00
 -0.289564E-02  0.447560E-02


KBIT TRANSPOSED

MI=  0
  1  0.178738E+01 -0.478272E-02
  3 -0.143319E+01  0.222509E-02
  5  0.615180E+00 -0.889499E-03

MI=  1
  2  0.214443E+01 -0.522181E-02
  4 -0.923655E+00  0.773299E-03

MI=  2
  3  0.228042E+01 -0.522024E-02
  5 -0.583817E+00  0.652401E-04

MI=  3
  4  0.233447E+01 -0.512674E-02

MI=  4
  5  0.235187E+01 -0.501683E-02


KIBT

MI=  0
  1 -0.779531E-03 -0.271902E-01
  3  0.139014E-03  0.211242E-01
  5 -0.165599E-02 -0.749874E-02

MI=  1
  2 -0.713678E-03 -0.324670E-01
  4  0.434461E-03  0.128167E-01
```

```
MI= 2
  3 -0.728343E-03 -0.343192E-01
  5  0.917905E-03  0.718743E-02

MI= 3
  4 -0.824948E-03 -0.349029E-01

MI= 4
  5 -0.962311E-03 -0.349330E-01


KNM

MI= 0
  1  0.636620E+00
  3  0.282942E+00
  5  0.181083E+00

MI= 1
  2  0.424413E+00
  4  0.226354E+00

MI= 2
  3  0.339531E+00
  5  0.194017E+00

MI= 3
  4  0.291026E+00

MI= 4
  5  0.258690E+00


BMNJ

MI= 0
  1  0.150000E+01 -0.572822E+00  0.359035E+00
  3 -0.572822E+00  0.204167E+01 -0.936910E+00
  5  0.359035E+00 -0.936910E+00  0.233177E+01

MI= 1
  2  0.187500E+01 -0.765466E+00
  4 -0.765466E+00  0.222656E+01

MI= 2
  3  0.218750E+01 -0.906890E+00
  5 -0.906890E+00  0.240625E+01

MI= 3
  4  0.246094E+01

MI= 4
```

```
 5  0.270703E+01

DKII

MI=  0
MJ=  0
 1  0.644836E-01 -0.300000E-01  0.119928E-01
 3 -0.300000E-01  0.862652E-01 -0.500630E-01
 5  0.119928E-01 -0.500630E-01  0.974699E-01

MI=  0
MJ=  1
 1  0.704036E-01 -0.104261E-01
 3 -0.553783E-01  0.851911E-01
 5  0.237267E-01 -0.761840E-01

MI=  0
MJ=  2
 1  0.703825E-01 -0.879611E-03
 3 -0.723492E-01  0.748906E-01
 5  0.374545E-01 -0.928213E-01

MI=  0
MJ=  3
 1  0.691218E-01
 3 -0.834695E-01
 5  0.507615E-01

MI=  0
MJ=  4
 1  0.676400E-01
 3 -0.908882E-01
 5  0.626552E-01

MI=  1
MJ=  1
 2  0.841232E-01 -0.352490E-01
 4 -0.352490E-01  0.100299E+00

MI=  1
MJ=  2
 2  0.892359E-01 -0.219973E-01
 4 -0.553842E-01  0.100025E+00

MI=  1
MJ=  3
 2  0.912058E-01
 4 -0.705437E-01

MI=  1
MJ=  4
```

```
  2  0.917869E-01
  4 -0.819009E-01

MI=  2
MJ=  2
  3  0.985135E-01 -0.417552E-01
  5 -0.417552E-01  0.109053E+00

MI=  2
MJ=  3
  3  0.103550E+00
  5 -0.581713E-01

MI=  2
MJ=  4
  3  0.106373E+00
  5 -0.714583E-01

MI=  3
MJ=  3
  4  0.111103E+00

MI=  3
MJ=  4
  4  0.115934E+00

MI=  4
MJ=  4
  5  0.122473E+00


RB
  0.138400E-03  0.971509E-03  0.108864E+00
 -0.636800E-05 -0.268670E-04 -0.232424E-02


RIT

MI=  0
  1  0.000000E+00  0.324546E-04  0.361573E-02
  3  0.000000E+00 -0.191803E-04 -0.220228E-02
  5  0.000000E+00 -0.260126E-04 -0.284153E-02

MI=  1
  2  0.000000E+00  0.381387E-04  0.426726E-02
  4  0.000000E+00  0.195088E-05  0.132647E-03

MI=  2
  3  0.000000E+00  0.376589E-04  0.423138E-02
  5  0.000000E+00  0.192015E-04  0.206061E-02

MI=  3
```

```
  4  0.000000E+00  0.345722E-04  0.390058E-02

MI=  4
  5  0.000000E+00  0.305127E-04  0.345639E-02


COB
  0.819502E-02
  0.223154E-03


COIT

MI=  0
  1  0.471127E-03
  3  0.342787E-03
  5 -0.958044E-04

MI=  1
  2  0.331479E-03
  4  0.491536E-03

MI=  2
  3  0.206420E-03
  5  0.460440E-03

MI=  3
  4  0.121108E-03

MI=  4
  5  0.647152E-04
```

Trim Calculation:

File 'trim.f'

```
      SUBROUTINE TRIM(MATFILE,NCAP,DIM,DIMIN,DIMINS,KMAX,V,VT,SMT)
C*****************************************************************
C
C     CALCULATION OF TRIM STATE
C     READING MATRIX PARTS FROM FILE "MATFILE" AND ASSEMBLING THE
C     SYSTEM STIFFNESS MATRIX "KSYS", CONTROL MATRIX "RSYS" AND
C     CONSTANT VECTOR "COSYS" IN THE SUBROUTINES
C     BBSORT, NISORT, IBSORT, DKIIMAT, AND KIIMAT.
C     (REM: ONLY 3RD COLUMN OF "RSYS" IS NEEDED
C        -CONTROL INPUT TIME DERIVATIVE CONTRIBUTIONS SKIPPED)
C
C     THE RESULT IS WRITTEN IN THE FILE "TRIMFILE".
C
C     NCAP  -     HIGHEST ORDER OF RADIAL POLYNOMIAL EXPANSION
C     DIM   -     DIMENSION OF COMPLETE SYSTEM MATRICES
C     DIMIN-      DIMENSION OF INFLOW PARTITION OF COMPLETE SYSTEM
C     DIMINS-     DTO., SIN-PARTITION ONLY (Bnm COEFFICIENTS)
C     KMAX-       MAX. HARMONICS COUNTER
C     VT    -     MASS FLOW PARAMETER VT=Vinf+nu
C     V     -     EQUIVALENT MASS FLOW PARAMETER V=Vinf+2*nu
C     SMT   -     SIMPLE MOMENTUM THEORY VALUE FOR a01
C                 (I.E. CONSTANT INFLOW DISTRIBUTION)
C
C     FILE UNIT NUMERING USED:
C           7     -     INPUT FILE (MATFILE IN SORT SUBROUTINES)
C           8     -     OUTPUT FILE (TRIMFILE IN THIS SUBROUTINE)
C
C     THIS VERSION USES LU DECOMPOSITION (ROUTINES LUDCMP AND LUBKSB)
C
C*****************************************************************
      INTEGER NCAP,DIM,DIMJ,DIMIN,DIMINS, KMAX
      INTEGER MAXIT,I,K,J, Q,INDX(DIM),D,SING
      REAL  KSYS(DIM,DIM),RSYS(DIM,3),COSYS(DIM,1),B(DIM),VT,V,SMT
      REAL DELTAC,DELTAC1,DELTAC2,DELTACL,DELTACH,DDELTAC
      REAL EPS,EPS1,FL,FH,F,SWAP,DEL
      CHARACTER*8 MATFILE, TRIMFILE
C
C
      PRINT*,'MODULE TRIM'
      PRINT*
C
      DO 2 I=1,DIM
        COSYS(I,1)=0.
         RSYS(I,1)=0.
         RSYS(I,2)=0.
```

```
          RSYS(I,3)=0.
          DO 1 J=1,DIM
               KSYS(I,J)=0.
 1        CONTINUE
 2      CONTINUE
C
C       ITERATION PARAMETER
C
        EPS1=1.E-8
        EPS=1.E-8
        MAXIT=10
        DELTAC1=-0.5
        DELTAC2=0.1
C
C       SYSTEM STIFFNESS MATRIX KSYS
C
        DIMJ=DIM
        CALL BBSORT(MATFILE,KSYS,'KBB ',DIM,DIMJ,2,0)
        CALL BISORT(MATFILE,KSYS,'KBIT',DIM,2,1,0,KMAX,NCAP,DIMINS,DIMIN)
        CALL IBSORT(MATFILE,KSYS,'KIBT',DIM,DIMJ,2,1,0,KMAX,NCAP,
     $                  DIMINS,DIMIN,0)
        CALL KIIMAT (MATFILE,KSYS,DIM,1,0,KMAX,NCAP,DIMINS,DIMIN,V,VT)
        CALL DKIIMAT(MATFILE,KSYS, DIM,1,0,KMAX,NCAP,DIMINS,DIMIN)
        PRINT*,'      TRIM: ASSEMBLED KSYS'
C
C       SYSTEM CONTROL MATRIX RSYS
C
        CALL BBSORT(MATFILE,RSYS,'RB  ',DIM,3,3,1)
        CALL IBSORT(MATFILE,RSYS,'RIT ',DIM,3,3,1,0,KMAX,NCAP,
     $                  DIMINS,DIMIN,1)
        PRINT*,'      TRIM: ASSEMBLED RSYS'
C
C       SYSTEM CONSTANT VECTOR COSYS
C
        CALL BBSORT(MATFILE,COSYS,'COB ',DIM,1,1,1)
        CALL IBSORT(MATFILE,COSYS,'COIT',DIM,1,1,1,0,KMAX,NCAP,
     $                  DIMINS,DIMIN,1)
        PRINT*,'      TRIM: ASSEMBLED COSYS'
C
C       TRIM ITERATION
C
        PRINT*,'      TRIM: STARTING TRIM ITERATION'
        DO 10 I=1,DIM
          B(I)=RSYS(I,3)*DELTAC1+COSYS(I,1)
 10     CONTINUE
        CALL LUDCMP(KSYS,DIM,DIM,INDX,D,SING)
        IF(SING.EQ.1) STOP
        CALL LUBKSB(KSYS,DIM,DIM,INDX,B)
        FL=B(9)-SMT
C
        DO 20 I=1,DIM
          B(I)=RSYS(I,3)*DELTAC2+COSYS(I,1)
```

```
 20     CONTINUE
        CALL LUBKSB(KSYS,DIM,DIM,INDX,B)
        FH=B(9)-SMT
C
C
        IF (FL.LT.0.) THEN
              DELTACL=DELTAC1
              DELTACH=DELTAC2
        ELSE
              DELTACL=DELTAC2
              DELTACH=DELTAC1
              SWAP=FL
              FL=FH
              FH=SWAP
        ENDIF
        DDELTAC=DELTACH-DELTACL
        DO 35 J=1,MAXIT
              DELTAC=DELTACL+DDELTAC *FL/(FL-FH)
C
              DO 30 I=1,DIM
                 B(I)=RSYS(I,3)*DELTAC+COSYS(I,1)
 30           CONTINUE
              CALL LUBKSB(KSYS,DIM,DIM,INDX,B)
              F=B(9)-SMT
C
              IF (F.LT.0.) THEN
                    DEL=DELTACL-DELTAC
                    DELTACL=DELTAC
                    FL=F
              ELSE
                    DEL=DELTACH-DELTAC
                    DELTACH=DELTAC
                    FH=F
              ENDIF
              DDELTAC=DELTACH-DELTACL
              PRINT*,'      TRIM: FINISHED ITERATION ',J,'; DELTAC=',DELTAC
              IF((ABS(DEL).LT.EPS).OR.(F.EQ.0.)) GOTO 40
 35     CONTINUE
        PAUSE 'TRIM exceeded maximum iterations!'
 40     CONTINUE
        PRINT*,'      TRIM: FOUND TRIM STATE'
C
C       WRITE RESULT IN "TRIMFILE"
C
        TRIMFILE(:4)=MATFILE(:4)
        TRIMFILE(5:)=CHAR(NCAP/10+48)
        TRIMFILE(6:)=CHAR(NCAP-(NCAP/10)*10+48)
        TRIMFILE(7:)='TR'
        OPEN(UNIT=8,FILE=TRIMFILE)
        WRITE(8,500)DELTAC
        DO 43 Q=1,4
           WRITE(8,510) Q, B(2*(Q-1)+1)
```

```
          WRITE(8,520) Q, B(2*(Q-1)+2)
 43     CONTINUE
        I=9
        DO 50 M=0,NCAP
          JMAX=(NCAP-M)/2
          DO 45 J=0,JMAX
               N=M+1+2*J
               IF(B(I).LT.EPS1)  B(I)=0.
               WRITE(8,530)M,N,B(I)
               I=I+1
 45       CONTINUE
 50     CONTINUE
        DO 60 M=1,NCAP
          JMAX=(NCAP-M)/2
          DO 55 J=0,JMAX
               N=M+1+2*J
               IF(B(I).LT.EPS1)  B(I)=0.
               WRITE(8,540)M,N,B(I)
               I=I+1
 55       CONTINUE
 60     CONTINUE
        CLOSE(8)
        PRINT*,'      TRIM: TRIM STATE WRITTEN IN FILE ',TRIMFILE
C
 500    FORMAT(//,10X, 'DELTAC          ',E14.6,/)
 510    FORMAT(10X,'BETA ',I3,'         ',E14.6)
 520    FORMAT(10X,'THETA',I3,'         ',E14.6)
 530    FORMAT(10X,'AMN',I2,X,I2,'      ',E14.6)
 540    FORMAT(10X,'BMN',I2,X,I2,'      ',E14.6)
C
        RETURN
        END


        SUBROUTINE LUBKSB(A,N,NP,INDX,B)
C*****************************************************************
C
C       SOLVES A SET OF N LINEAR EQUATIONS A.X=B. HERE A IS INPUT, NOT
C       AS THE MATRIX A BUT RATHER THE LUI DECOMPOSITION, DETERMINED
C       BY THE ROUTINE LUDCMP. B IS INPUT OF THE RIGHT-HAND SIDE VECTOR B,
C       AND RETURNS WITH THE SOLUTION VECTOR X. A, N, NP AND INDX ARE NOT
C       MODIFIED BY THIS ROUTINE AND CAN BE LEFT IN PLACE FOR SUCCESSIVE
C       CALLS WITH DIFFERENT RIGHT-HAND SIDES B. THIS ROUTINE TAKES INTO
C       ACCOUNT THE POSSIBILTY THAT B WILL BEGIN WITH MANY ZERO
C       ELEMENTS, SO IT IS EFFICIENT FOR USE IN MATRIX INVERSION.
C
C       SOURCE:       "NUMERICAL RECIPES"
C
C       VERSION NOVEMBER 27, 1990
C
C       MARTIN STETTNER
C
```

```
C*****************************************************************
      DIMENSION A(NP,NP), INDX(N),B(N)
      II=0
      DO 12 I=1,N
        LL=INDX(I)
        SUM=B(LL)
        B(LL)=B(I)
        IF(II.NE.0) THEN
            DO 11 J=II,I-1
               SUM=SUM-A(I,J)*B(J)
11          CONTINUE
        ELSE IF (SUM.NE.0) THEN
            II=I
        END IF
        B(I)=SUM
12    CONTINUE
      DO 14 I=N,1,-1
        SUM=B(I)
        DO 13 J=I+1,N
           SUM=SUM-A(I,J)*B(J)
13      CONTINUE
        B(I)=SUM/A(I,I)
14    CONTINUE
      RETURN
      END

      SUBROUTINE LUDCMP(A,N,NP,INDX,D,ERR)
C*****************************************************************
C
C     GIVEN A NxN MATRIX A, WITH PHYSICAL DIMENSION NP, THIS ROUTINE
C     REPLACES IT BY THE LU DECOMPOSITION OF A ROWWISE PERMUTATION OF
C     ITSELF. A AND N ARE INPUT. A IS OUTPUT, ARRANGED IN UPPER AND
C     LOWER TRIDIAGONAL MATRIX (AS IN EQ. 2.3.14 OF "NUMERICAL
C     RECIPES"); INDX IS AN OUTPUT VECTOR WHICH RECORDS THE ROW
C     PERMUTATION BY THE PARTIAL PIVOTING; D IS OUTPUT AS +/- 1
C     DEPENDING ON WETHER THE NUMBER OF ROWS EXCHANGED WAS EVEN OR
C     ODD, RESPECTIVELY. THIS ROUTINE IS USED IN COMBINATION WITH
C     LUBKSB TO SOLVE LINEAR EQUATIONS OR INVERT A MATRIX.
C
C     SOURCE:       "NUMERICAL RECIPES"
C
C     VERSION NOVEMBER 27, 1990
C
C     MARTIN STETTNER
C
C*****************************************************************
      PARAMETER (NMAX=100,TINY=1.0E-20)
      DIMENSION A(NP,NP), INDX(N), VV(NMAX)
C
      ERR=0
      D=1.
      DO 12 I=1,N
```

```
      AAMAX=0.
      DO 11 J=1,N
            IF(ABS(A(I,J)).GT.AAMAX)  AAMAX=ABS(A(I,J))
11    CONTINUE
      IF(AAMAX.EQ.0.)THEN
            PRINT*,'              LUDCMP: SINGULAR MATRIX'
            ERR=1
            RETURN
      END IF
       VV(I)=1./AAMAX
12    CONTINUE
      DO 19 J=1,N
        DO 14 I=1,J-1
            SUM=A(I,J)
            DO 13 K=1,I-1
               SUM=SUM-A(I,K)*A(K,J)
13          CONTINUE
            A(I,J)=SUM
14    CONTINUE
      AAMAX=0.
      DO 16 I=J,N
            SUM=A(I,J)
            DO 15 K=1,J-1
               SUM=SUM-A(I,K)*A(K,J)
15          CONTINUE
            A(I,J)=SUM
            DUM=VV(I)*ABS(SUM)
            IF(DUM.GE.AAMAX) THEN
              IMAX=I
              AAMAX=DUM
            END IF
16    CONTINUE
      IF(J.NE.IMAX) THEN
            DO 17 K=1,N
              DUM=A(IMAX,K)
               A(IMAX,K)=A(J,K)
              A(J,K)=DUM
17          CONTINUE
            D=-D
            VV(IMAX)=VV(J)
      END IF
      INDX(J)=IMAX
       IF(A(J,J).EQ.0.)  A(J,J)=TINY
      IF(J.NE.N) THEN
            DUM=1./A(J,J)
            DO 18 I=J+1,N
               A(I,J)=A(I,J)*DUM
18          CONTINUE
        END IF
19    CONTINUE
      RETURN
      END
```

Sample Output:

Trim States for four Radial Expansion Functions

(N = 4)

K10104TR

DELTAC	-0.232804E-01
BETA 1	0.443892E-01
THETA 1	0.131199E+00
BETA 2	0.443892E-01
THETA 2	0.131199E+00
BETA 3	0.443892E-01
THETA 3	0.131199E+00
BETA 4	0.443892E-01
THETA 4	0.131199E+00
AMN 0 1	0.346410E-01
AMN 0 3	0.000000E+00
AMN 0 5	0.000000E+00
AMN 1 2	0.000000E+00
AMN 1 4	0.000000E+00
AMN 2 3	0.000000E+00
AMN 2 5	0.000000E+00
AMN 3 4	0.000000E+00
AMN 4 5	0.339019E-02
BMN 1 2	0.000000E+00
BMN 1 4	0.000000E+00
BMN 2 3	0.000000E+00
BMN 2 5	0.000000E+00
BMN 3 4	0.000000E+00
BMN 4 5	0.000000E+00

Frequency Response:

File 'fresp.f'

```
      SUBROUTINE FRESP(MATFILE,NCAP,MODEL,DIM,DIMINM,DIMINSM,
     $                 DM,M0,KMAX,V, NRMIN,NRMAX,DNR,SWITCH)
C*****************************************************************
C
C     FREQUENCY SWEEP FOR LINEAR BLADE DYNAMICS/
C     INFLOW DYNAMICS MODEL
C     MATRICES ARE ASSEMBLED ACCORDING TO THE TYPE OF INPUT CHOSEN
C     MATRIX PARTITIONS ARE READ FROM "MATFILE" AND ASSEMBLED
C     SYSTEM IS SOLVED FOR REAL AND IMAGINARY PART OF THE STATE
C     VARIABLES BY THE ROUTINES "LUDCMP" AND "LUBKSB"
C     (LOCATED IN FILE trim.f)
C     RESULTS ARE WRITTEN IN THE FILE "PLOTFILE"
C
C     MATFILE-   FILE OF COMMON MATRIX PARTITIONS
C     NCAP -     MAXIMAL ORDER OF RADIAL SHAPE FUNCTIONS
C     MODEL      -    INPUT/MODEL TYPE:
C                1    ALL HARMONICS, ONE BLADE INPUT
C                2    COLLECTIVE INPUT (TRUNCATED MODEL)
C                3    PROGRESSING INPUT (TRUNCATED MODEL)
C                4    REGRESSING INPUT (TRUNCATED MODEL)
C                5    DIFFERENTIAL INPUT
C     DIM  -     DIMENSION OF SYSTEM MATRICES (NO. OF STATES)
C     DIMINM     -    NUMBER OF INFLOW STATES, TRUNCATED
C     DIMINSM    -    DTO., SIN PARTITION ONLY
C     DM   -     HARMONICS STEP
C     M0   -     SMALLEST HARMONIC INCLUDED/EXCITED
C     KMAX-      NUMBER OF HARMONICS EXCITED BY CONTROL INPUT (-1)
C     V    -     MASS FLOW PARAMETER; V=Vinf + nu
C     NR   -     NORMALIZED CONTROL INPUT FREQUENCY,
C                ROTATING SYSTEM
C     NRMIN-     LOWER LIMIT OF FREQUENCY SWEEP
C     NRMAX-     UPPER LIMIT OF FREQUENCY SWEEP
C     DNR  -     STEPWIDTH
C     SWITCH     -    OUTPUT MODE:
C                1    AMPLITUDE/PHASE
C                ELSE REAL/IMAGINARY PARTS
C
C*****************************************************************
      INTEGER I,I1,I2,IMAX,J,JMAX,K,Q,QMAXA,QMAXB,CUTS(8)
      INTEGER NCAP,DM,M0,KMAX
      INTEGER DIM,DIMINM,DIMINSM
      INTEGER MODEL,ROW,NI,SWITCH
      INTEGER NRLP,NRMINLP,NRMAXLP,DNRLP,D,SING,INDX(2*DIM)
```

```
      REAL V,SGN,ALT,RES
      REAL NR,NRMIN,NRMAX,DNR
      REAL MSYS(DIM,DIM),CSYS(DIM,DIM),KSYS(DIM,DIM)
      REAL RI(DIMINM,3),RB(2,3)
      REAL A(2*DIM,2*DIM),B(2*DIM)
      REAL BAMP(4),TAMP(4),BPHA(4),TPHA(4),PHAOLD(8)
      REAL DUM
      CHARACTER*3 DUMMY3
      CHARACTER*4 DUMMY4
      CHARACTER*8 DUMMY8,MATFILE
      CHARACTER*10 PLOTFILEA,PLOTFILEB
C
      EXTERNAL PHASE
      RES=100.
      IF(MODEL.EQ.1) THEN
        QMAXB=4
        QMAXA=2
      ELSE
        QMAXA=1
        QMAXB=1
      END IF
      DO 1 Q=1,8
        CUTS(J)=0
 1    CONTINUE
C
      PLOTFILEA(:4)=MATFILE(:4)
      PLOTFILEA(5:)=CHAR(NCAP/10+48)
      PLOTFILEA(6:)=CHAR(NCAP-(NCAP/10)*10+48)
      PLOTFILEA(7:)='PL'
      PLOTFILEA(9:)=CHAR(MODEL+48)
      PLOTFILEA(10:)='A'
      PLOTFILEB=PLOTFILEA
      PLOTFILEB(10:)='B'
C
      PRINT*
      PRINT*,'MODULE FRESP'
C
C     ASSEMBLY OF SYSTEM MASS MATRIX MSYS
C
      PRINT*
      PRINT*,'     FRESP: STARTING MATRIX ASSEMBLY, MODEL:', MODEL
      DO 10 I=1,DIM
        DO 5 J=1,DIM
          MSYS(I,J)=0.
 5      CONTINUE
 10   CONTINUE
      CALL BBSORT (MATFILE,MSYS,'MBB ',DIM,DIM,2,0)
      CALL IBSORT (MATFILE,MSYS,'MIBT',DIM,DIM,2,DM,M0,KMAX,NCAP,
     $                  DIMINSM,DIMINM,0)
      PRINT*,'     FRESP: ASSEMBLED MSYS'
C
C     ASSEMBLY OF SYSTEM DAMPING MATRIX CSYS
```

```
C
      DO 20 I=1,DIM
        DO 15 J=1,DIM
             CSYS(I,J)=0.
15      CONTINUE
20    CONTINUE
      CALL BBSORT (MATFILE,CSYS,'CBB ',DIM,DIM,2,0)
      CALL IBSORT (MATFILE,CSYS,'CIBT',DIM,DIM,2,DM,M0,KMAX,NCAP,
     $                    DIMINSM,DIMINM,0)
      CALL KNMDIAG(MATFILE,CSYS,DIM,DM,M0,KMAX,NCAP,
     $                      DIMINSM,DIMINM)
      PRINT*,'      FRESP: ASSEMBLED CSYS'
C
C     ASSEMBLY OF SYSTEM STIFFNESS MATRIX KSYS
C
      DO 30 I=1,DIM
        DO 25 J=1,DIM
             KSYS(I,J)=0.
25      CONTINUE
30    CONTINUE
      CALL BBSORT(MATFILE,KSYS,'KBB ',DIM,DIM,2,0)
      CALL BISORT(MATFILE,KSYS,'KBIT',DIM,2,DM,M0,KMAX,NCAP,
     $              DIMINSM,DIMINM)
      CALL IBSORT(MATFILE,KSYS,'KIBT',DIM,DIM,2,DM,M0,KMAX,NCAP,
     $              DIMINSM,DIMINM,0)
      CALL KIIMAT  (MATFILE,KSYS,DIM,DM,M0,KMAX,NCAP,
     $              DIMINSM,DIMINM,V,V)
      CALL DKIIMAT(MATFILE,KSYS,DIM,DM,M0,KMAX,NCAP,
     $              DIMINSM,DIMINM)
      PRINT*,'      FRESP: ASSEMBLED KSYS'
C
C     READING IN CONTROL MATRIX PARTITIONS FROM MATFILE
C
      OPEN (UNIT=7,FILE=MATFILE)
      READ(7,*)  DUMMY8
      IF (DUMMY8.NE.MATFILE) THEN
        PRINT*,'ERROR WHILE ATTEMPTING TO READ MATFILE:'
        PRINT*,'FILENAME AND FILE CODE ARE NOT EQUIVALENT'
        STOP
      ELSE
      END IF
C
35    CONTINUE
      READ(7,400)DUMMY4
      IF(DUMMY4.NE.'RB  ') GOTO 35
      DO 40 I=1,2
         READ(7,520)  (RB(I,J),J=1,3)
40    CONTINUE
45    CONTINUE
      READ(7,400)DUMMY4
      IF(DUMMY4.NE.'RIT ') GOTO 45
      READ(7,400)DUMMY4
```

```
      ROW=0
      DO 60 K=0,KMAX
        M=DM*K+M0
50      CONTINUE
        READ(7,500)DUMMY3,SEARCHM
        SEARCHM=SEARCHM*10**6
        IF((DUMMY3.NE.'MI=').OR.(SEARCHM.NE.REAL(M))) GOTO 50
        IMAX=(NCAP-M)/2
        DO 55 I=0,IMAX
            ROW=ROW+1
            READ(7,510)NI,(RI(ROW,J),J=1,3)
55      CONTINUE
60    CONTINUE
      CLOSE(7)
      PRINT*,'      FRESP: COMPLETED READ-IN PARTITIONS OF RSYS'
      PHAOLD(1) = -180.
      PHAOLD(2) = -180.
      CUTS(1)=0
      CUTS(2)=0
      IF(MODEL.EQ.1) THEN
        DO 65 Q=2,QMAXB
            CUTS(2*(Q-1)+1)=0
            CUTS(2*(Q-1)+2)=0
            PHAOLD(2*(Q-1)+1) = 0.
            PHAOLD(2*(Q-1)+2) = 0.
65      CONTINUE
      END IF
C*****************************************************************
C
C     BEGIN FREQUENCY SWEEP
C
C*****************************************************************
      OPEN(UNIT=8,FILE=PLOTFILEA)
      WRITE(8,*) PLOTFILEA
      IF(MODEL.EQ.1) THEN
        OPEN(UNIT=9,FILE=PLOTFILEB)
        WRITE(9,*) PLOTFILEB
      END IF
      NRMINLP=INT(NRMIN*RES)
      NRMAXLP=INT(NRMAX*RES)
      DNRLP  =INT(DNR  *RES)
      DO 1000 NRLP=NRMINLP,NRMAXLP,DNRLP
        NR=REAL(NRLP)/RES
        DO 75 I=1,2*DIM
            DO 74 J=1,2*DIM
              A(I,J)=0.
74          CONTINUE
75      CONTINUE
C
C       FEED REAL PART OF LHS MATRICES INTO MATRIX A
C
        DO 110 I=1,DIM
```

```
            DO 105 J=1,DIM
                A(I,J)=KSYS(I,J)-NR*NR*MSYS(I,J)
                A(DIM+I,DIM+J)=A(I,J)
105         CONTINUE
110     CONTINUE
C
C       FEED IMAGINARY PART OF LHS MATRICES INTO MATRIX A
C
        DO 120 I=1,DIM
            DO 115 J=1,DIM
                A(DIM+I,J)=NR*CSYS(I,J)
                A(I,DIM+J)=(-1)*A(DIM+I,J)
115         CONTINUE
120     CONTINUE
C
C       FEED REAL PART OF RHS VECTOR INTO B
C       USING TRIG. IDENTITIES
C
        IF (MODEL.EQ.1) THEN
            DO 125 I=1,2
                B(I)=RB(I,3)-NR*NR*RB(I,1)
125         CONTINUE
            DO 130 I=3,8
                B(I)=0.
130         CONTINUE
            DO 135 I=9,DIM-DIMINSM
                B(I)=RI(I-8,3)-NR*NR*RI(I-8,1)
135         CONTINUE
            DO 136 I=DIM-DIMINSM+1,DIM
                B(I)=RI(I-8-DIMINSM,3)
     $                  -NR*NR*RI(I-8-DIMINSM,1)
136         CONTINUE
        ELSE IF(MODEL.EQ.2) THEN
            DO 145 Q=1,4
              DO 140 I=1,2
                  B(2*(Q-1)+I)=RB(I,3)-NR*NR*RB(I,1)
140           CONTINUE
145         CONTINUE
            IMAX=8+NCAP/2+1
            DO 150 I=8,IMAX
                B(I)=2*RI(I-8,3)
150         CONTINUE
            DO 155 I=IMAX+1,DIM-DIMINSM
                B(I)=4*RI(I-8,3)
155         CONTINUE
            DO 160 I=DIM-DIMINSM+1,DIM
              B(I)=0.
160         CONTINUE
        ELSE IF ((MODEL.EQ.3).OR.(MODEL.EQ.4)) THEN
            IF (MODEL.EQ.4) THEN
              SGN=-1.
            ELSE
```

```
                SGN=1.
              END IF
              DO 165 I=1,2
                  B(I)=    RB(I,3)-NR*NR*RB(I,1)
                 B(I+4)=-B(I)
                 B(I+2)= SGN*NR*RB(I,2)
                 B(I+6)=-B(I+2)
165           CONTINUE
              DO 170 I=9,DIM-DIMINSM
                  B(I)=2.*RI(I-8,3)
170           CONTINUE
              I=DIM-DIMINSM
              ALT=-1.
              DO 180 K=0,(NCAP-1)/2
                 ALT=ALT*(-1.)
                M=2*K+1
                 JMAX=(NCAP-M)/2
                DO 175 J=0,JMAX
                     I=I+1
                     B(I)=ALT*SGN*2*NR*RI(I-8-DIMINSM,2)
175             CONTINUE
180           CONTINUE
            ELSE
              DO 185 I=1,2
                  B(I)=    RB(I,3)-NR*NR*RB(I,1)
                 B(I+2)=-B(I)
                 B(I+4)= B(I)
                 B(I+6)=-B(I)
185           CONTINUE
              DO 190 I=9,DIM-DIMINSM
                  B(I)=4.*RI(I-8,3)
190           CONTINUE
              DO 200 I=DIM-DIMINSM+1,DIM
                 B(I)=0.
200           CONTINUE
            END IF
C
C           FEED IMAGINARY PART OF RHS VECTOR INTO B
C            USING TRIG. IDENTITIES
C
            IF (MODEL.EQ.1) THEN
              DO 225 I=1,2
                  B(DIM+I)=NR*RB(I,2)
225           CONTINUE
              DO 230 I=3,8
                 B(DIM+I)=0.
230           CONTINUE
              DO 235 I=9,DIM-DIMINSM
                  B(DIM+I)=NR*RI(I-8,2)
235           CONTINUE
              DO 236 I=DIM-DIMINSM+1,DIM
                  B(DIM+I)=NR*RI(I-8-DIMINSM,2)
```

```
236         CONTINUE
        ELSE IF (MODEL.EQ.2) THEN
            DO 245 Q=1,4
              DO 240 I=1,2
                  B(DIM+2*(Q-1)+I)=NR*RB(I,2)
240           CONTINUE
245         CONTINUE
            IMAX=8+NCAP/2+1
            DO 250 I=8,IMAX
                B(DIM+I)=2*NR*RI(I-8,2)
250         CONTINUE
            DO 255 I=IMAX+1,DIM-DIMINSM
                B(DIM+I)=4*NR*RI(I-8,2)
255         CONTINUE
            DO 260 I=DIM-DIMINSM+1,DIM
              B(DIM+I)=0.
260         CONTINUE
        ELSE IF ((MODEL.EQ.3).OR.(MODEL.EQ.4)) THEN
            IF (MODEL.EQ.4) THEN
              SGN=-1.
            ELSE
              SGN=1.
            END IF
            DO 265 I=1,2
               B(DIM+I)=   NR*RB(I,2)
               B(DIM+I+4)=-B(DIM+I)
                B(DIM+I+2)=-SGN*(RB(I,3)-NR*NR*RB(I,1))
               B(DIM+I+6)=-B(DIM+I+2)
265         CONTINUE
            DO 270 I=9,DIM-DIMINSM
                B(DIM+I)=2*NR*RI(I-8,2)
270         CONTINUE
            I=DIM-DIMINSM
            ALT=1.
            DO 280 K=0,(NCAP-1)/2
              ALT=ALT*(-1.)
              M=2*K+1
              JMAX=(NCAP-M)/2
              DO 275 J=0,JMAX
                  I=I+1
                  B(DIM+I)=ALT*SGN*2*RI(I-8-DIMINSM,3)
275           CONTINUE
280         CONTINUE
        ELSE
            DO 285 I=1,2
               B(I+DIM)=   NR*RB(I,2)
               B(I+DIM+2)=-B(I+DIM)
               B(I+DIM+4)= B(I+DIM)
               B(I+DIM+6)=-B(I+DIM)
285         CONTINUE
            DO 290 I=9,DIM-DIMINSM
                B(I+DIM)=4*NR*RI(I-8,2)
```

```
290         CONTINUE
            DO 300 I=DIM-DIMINSM+1,DIM
               B(I+DIM)=0.
300         CONTINUE
         END IF
C
C        SOLVE FOR STATE VARIABLE REAL AND IMAGINARY PART,
C        CALCULATE PHASE AND AMPLITUDE OF BETA AND THETA
C
          CALL LUDCMP(A,2*DIM,2*DIM,INDX,D,SING)
         IF(SING.EQ.1) THEN
            PRINT*,'     FRESP: NR= ',NR,' SKIPPED'
            GOTO 360
         END IF
          CALL LUBKSB(A,2*DIM,2*DIM,INDX,B)
          PRINT*,'   FRESP: CALCULATED RESPONSE AT NR= ',NR
         IF(SWITCH.EQ.1) THEN
            DO 350 Q=1,QMAXB
              I1=2*(Q-1)+1
              I2=I1+DIM
               BAMP(Q)=SQRT(B(I1)*B(I1)+B(I2)*B(I2))
               TAMP(Q)=SQRT(B(I1+1)*B(I1+1)+B(I2+1)*B(I2+1))
               BPHA(Q)=PHASE(B(I2),B(I1),PHAOLD(I1),CUTS(I1))
               TPHA(Q)=PHASE(B(I2+1),B(I1+1),PHAOLD(I1+1),CUTS(I1+1))
              PHAOLD(I1)=BPHA(Q)
              PHAOLD(I1+1)=TPHA(Q)
350         CONTINUE
            WRITE(8,600)NR,((BAMP(Q),BPHA(Q),
     $                  TAMP(Q),TPHA(Q)),Q=1,QMAXA)
            IF(MODEL.EQ.1) THEN
               WRITE(9,600)NR,((BAMP(Q),BPHA(Q),
     $                  TAMP(Q),TPHA(Q)),Q=3,QMAXB)
            END IF
         ELSE
            WRITE(8,700)NR,((B(2*(Q-1)+1),B(2*(Q-1)+1+DIM),
     $                  B(2*(Q-1)+2),B(2*(Q-1)+2+DIM)),Q=1,QMAXA)
            IF(MODEL.EQ.1) THEN
               WRITE(9,700)NR,((B(2*(Q-1)+1),B(2*(Q-1)+1+DIM),
     $                     B(2*(Q-1)+2),B(2*(Q-1)+2+DIM)),Q=3,QMAXB)
            END IF
         END IF
360      CONTINUE
          IF(NR.EQ.1.1)THEN
            OPEN(UNIT=7,FILE='aedump2')
            DO 361 I = 1,DIM
                  WRITE(7,'(2E12.5)')B(I),B(I+DIM)
361         CONTINUE
            CLOSE (7)
          END IF
1000   CONTINUE
       CLOSE(8)
       IF(MODEL.EQ.1) CLOSE(9)
```

```
      PRINT*
      PRINT*,'     FRESP: BLADE RESPONSE WRITTEN IN FILES ',
     $             PLOTFILEA,', ',PLOTFILEB
C
 400  FORMAT(A4)
 500  FORMAT(A3,E14.6)
 510  FORMAT(I3,3E14.6)
 520  FORMAT(3E14.6)
 600  FORMAT(F3.1,4(E10.3,F8.1))
 700  FORMAT(F3.1,8F9.5)
C
      RETURN
      END


      REAL FUNCTION PHASE (I,R,PHOLD,C)
C*****************************************************************
C
C     RETURNS PHASE ANGLE OF FREQUENCY RESPONSE IN DEGREES
C     CALCULATES TOTAL PHASE SHIFT WITHOUT BRANCHCUTS WITHIN
C     +/- 180 DEGREES AROUND THE BLADE AZIMUTH IN THE ROTATING
C     SYSTEM PSIQ
C
C*****************************************************************
      REAL R, I, PI, DPHI,PH,PHOLD
      INTEGER C
C
      PI=4.*ATAN(1.)
      DPHI=20.
      IF ((I.EQ.0.).AND.(R.EQ.0.)) THEN
        PH=0.
      ELSE
         PH=180./PI*ATAN2(I,R)+C*360.
      END IF

C
      IF((PH-PHOLD).GT.(180.+DPHI)) THEN
        C=C-1
        PH = PH-360.
      ELSE IF ((PH-PHOLD).LT.-(180.+DPHI)) THEN
        C=C+1
        PH = PH+360.
      END IF
      PHASE=PH
      RETURN
      END
```

Sample Output:

Frequency Sweep for four Radial Expansion Functions,

Case: Deflection of one Servo Flap only

File A (of two)

K10104PL1A

0.0	0.491E+00	-180.0	0.898E+00	-180.0	0.165E-01	0.0	0.675E-03	0.0
0.1	0.498E+00	-185.8	0.901E+00	-183.1	0.162E-01	-26.3	0.105E-02	19.7
0.2	0.518E+00	-192.0	0.909E+00	-186.3	0.156E-01	-52.0	0.167E-02	6.7
0.3	0.553E+00	-199.0	0.923E+00	-189.6	0.148E-01	-77.0	0.227E-02	-14.9
0.4	0.604E+00	-207.0	0.943E+00	-193.2	0.143E-01	-102.6	0.281E-02	-39.0
0.5	0.674E+00	-216.4	0.968E+00	-197.3	0.141E-01	-130.5	0.331E-02	-64.7
0.6	0.769E+00	-228.0	0.995E+00	-201.9	0.141E-01	-163.5	0.372E-02	-92.8
0.7	0.888E+00	-242.7	0.102E+01	-207.3	0.137E-01	-204.6	0.395E-02	-123.9
0.8	0.101E+01	-261.5	0.103E+01	-213.1	0.113E-01	-251.5	0.402E-02	-159.5
0.9	0.110E+01	-284.4	0.103E+01	-218.6	0.930E-02	-268.7	0.396E-02	-212.2
1.0	0.109E+01	-309.7	0.103E+01	-222.7	0.196E-01	-309.5	0.306E-02	-302.7
1.1	0.969E+00	-333.3	0.107E+01	-227.1	0.262E-01	-377.7	0.274E-02	-412.3
1.2	0.829E+00	-353.7	0.113E+01	-233.6	0.256E-01	-436.4	0.254E-02	-484.8
1.3	0.698E+00	-372.0	0.118E+01	-242.2	0.220E-01	-484.2	0.151E-02	-523.0
1.4	0.578E+00	-388.9	0.120E+01	-252.3	0.176E-01	-524.0	0.985E-03	-487.2
1.5	0.470E+00	-404.5	0.117E+01	-263.1	0.132E-01	-557.4	0.221E-02	-485.9
1.6	0.375E+00	-418.4	0.110E+01	-273.6	0.937E-02	-584.4	0.328E-02	-512.3
1.7	0.295E+00	-430.3	0.101E+01	-283.2	0.638E-02	-604.6	0.382E-02	-539.5
1.8	0.232E+00	-440.3	0.910E+00	-291.5	0.431E-02	-616.8	0.391E-02	-564.0
1.9	0.183E+00	-448.4	0.810E+00	-298.6	0.306E-02	-621.1	0.372E-02	-585.4
2.0	0.146E+00	-455.1	0.719E+00	-304.5	0.246E-02	-620.6	0.339E-02	-603.9
2.1	0.118E+00	-460.4	0.639E+00	-309.4	0.221E-02	-620.7	0.302E-02	-619.8
2.2	0.962E-01	-464.9	0.569E+00	-313.6	0.210E-02	-623.8	0.266E-02	-633.6
2.3	0.796E-01	-468.5	0.509E+00	-317.0	0.201E-02	-629.3	0.233E-02	-645.6
2.4	0.667E-01	-471.6	0.457E+00	-320.0	0.191E-02	-636.2	0.204E-02	-656.2
2.5	0.565E-01	-474.3	0.413E+00	-322.6	0.179E-02	-643.6	0.178E-02	-665.5
2.6	0.484E-01	-476.5	0.374E+00	-324.8	0.166E-02	-651.0	0.156E-02	-673.9
2.7	0.418E-01	-478.5	0.341E+00	-326.7	0.153E-02	-658.1	0.137E-02	-681.3
2.8	0.365E-01	-480.2	0.312E+00	-328.4	0.140E-02	-664.9	0.121E-02	-687.9
2.9	0.320E-01	-481.7	0.286E+00	-329.9	0.128E-02	-671.2	0.107E-02	-693.9
3.0	0.284E-01	-483.1	0.263E+00	-331.2	0.117E-02	-677.1	0.946E-03	-699.2
3.1	0.253E-01	-484.3	0.243E+00	-332.4	0.106E-02	-682.6	0.843E-03	-704.1
3.2	0.226E-01	-485.4	0.225E+00	-333.5	0.970E-03	-687.7	0.754E-03	-708.4
3.3	0.204E-01	-486.4	0.210E+00	-334.5	0.888E-03	-692.4	0.678E-03	-712.3
3.4	0.185E-01	-487.3	0.195E+00	-335.4	0.814E-03	-696.8	0.613E-03	-715.9
3.5	0.168E-01	-488.2	0.182E+00	-336.2	0.749E-03	-701.0	0.556E-03	-719.2
3.6	0.154E-01	-489.0	0.171E+00	-336.9	0.691E-03	-705.1	0.508E-03	-722.3
3.7	0.141E-01	-489.8	0.160E+00	-337.6	0.639E-03	-708.9	0.465E-03	-725.2
3.8	0.130E-01	-490.5	0.150E+00	-338.2	0.592E-03	-712.7	0.428E-03	-728.0
3.9	0.120E-01	-491.3	0.141E+00	-338.8	0.550E-03	-716.4	0.396E-03	-730.6

4.0	0.112E-01	-492.0	0.133E+00	-339.3	0.512E-03	-720.0	0.367E-03	-733.2
4.1	0.104E-01	-492.6	0.126E+00	-339.8	0.476E-03	-723.5	0.340E-03	-735.8
4.2	0.968E-02	-493.3	0.119E+00	-340.2	0.444E-03	-726.9	0.317E-03	-738.3
4.3	0.905E-02	-493.9	0.113E+00	-340.6	0.414E-03	-730.3	0.295E-03	-740.8
4.4	0.849E-02	-494.6	0.107E+00	-341.0	0.386E-03	-733.6	0.276E-03	-743.2
4.5	0.797E-02	-495.2	0.102E+00	-341.4	0.360E-03	-736.8	0.258E-03	-745.6
4.6	0.750E-02	-495.8	0.965E-01	-341.7	0.336E-03	-739.9	0.241E-03	-747.9
4.7	0.707E-02	-496.5	0.919E-01	-342.0	0.313E-03	-742.9	0.226E-03	-750.2
4.8	0.667E-02	-497.1	0.875E-01	-342.3	0.292E-03	-745.9	0.211E-03	-752.5
4.9	0.631E-02	-497.7	0.835E-01	-342.6	0.273E-03	-748.7	0.198E-03	-754.7
5.0	0.598E-02	-498.2	0.797E-01	-342.8	0.255E-03	-751.4	0.186E-03	-756.8
5.1	0.567E-02	-498.8	0.761E-01	-343.0	0.238E-03	-754.1	0.174E-03	-758.9
5.2	0.538E-02	-499.4	0.728E-01	-343.3	0.222E-03	-756.6	0.163E-03	-760.9
5.3	0.512E-02	-499.9	0.697E-01	-343.5	0.208E-03	-759.1	0.153E-03	-762.8
5.4	0.487E-02	-500.5	0.667E-01	-343.6	0.194E-03	-761.4	0.144E-03	-764.6
5.5	0.464E-02	-501.0	0.639E-01	-343.8	0.182E-03	-763.6	0.135E-03	-766.4
5.6	0.443E-02	-501.5	0.613E-01	-344.0	0.170E-03	-765.8	0.127E-03	-768.1
5.7	0.422E-02	-502.1	0.589E-01	-344.1	0.159E-03	-767.8	0.120E-03	-769.8
5.8	0.404E-02	-502.6	0.565E-01	-344.2	0.149E-03	-769.8	0.113E-03	-771.4
5.9	0.386E-02	-503.0	0.543E-01	-344.4	0.140E-03	-771.7	0.106E-03	-772.9
6.0	0.370E-02	-503.5	0.522E-01	-344.5	0.131E-03	-773.5	0.999E-04	-774.3
6.1	0.354E-02	-504.0	0.503E-01	-344.6	0.123E-03	-775.3	0.942E-04	-775.7
6.2	0.339E-02	-504.4	0.484E-01	-344.7	0.116E-03	-776.9	0.889E-04	-777.1
6.3	0.326E-02	-504.9	0.466E-01	-344.8	0.109E-03	-778.5	0.839E-04	-778.3
6.4	0.313E-02	-505.3	0.449E-01	-344.9	0.102E-03	-780.0	0.793E-04	-779.6
6.5	0.300E-02	-505.7	0.433E-01	-344.9	0.965E-04	-781.5	0.750E-04	-780.7
6.6	0.289E-02	-506.1	0.417E-01	-345.0	0.910E-04	-782.9	0.710E-04	-781.9
6.7	0.278E-02	-506.5	0.403E-01	-345.1	0.858E-04	-784.2	0.672E-04	-782.9
6.8	0.268E-02	-506.9	0.389E-01	-345.1	0.810E-04	-785.5	0.637E-04	-784.0
6.9	0.258E-02	-507.2	0.376E-01	-345.2	0.765E-04	-786.7	0.604E-04	-784.9
7.0	0.248E-02	-507.6	0.363E-01	-345.2	0.723E-04	-787.9	0.573E-04	-785.9
7.1	0.240E-02	-507.9	0.351E-01	-345.2	0.684E-04	-789.0	0.544E-04	-786.8
7.2	0.231E-02	-508.3	0.339E-01	-345.3	0.648E-04	-790.1	0.517E-04	-787.7
7.3	0.223E-02	-508.6	0.328E-01	-345.3	0.614E-04	-791.1	0.491E-04	-788.5
7.4	0.216E-02	-508.9	0.317E-01	-345.3	0.582E-04	-792.1	0.467E-04	-789.3
7.5	0.208E-02	-509.2	0.307E-01	-345.3	0.552E-04	-793.0	0.445E-04	-790.0
7.6	0.202E-02	-509.5	0.297E-01	-345.3	0.525E-04	-794.0	0.424E-04	-790.8
7.7	0.195E-02	-509.8	0.288E-01	-345.3	0.498E-04	-794.8	0.404E-04	-791.5
7.8	0.189E-02	-510.1	0.279E-01	-345.3	0.474E-04	-795.7	0.385E-04	-792.1
7.9	0.183E-02	-510.4	0.270E-01	-345.3	0.451E-04	-796.5	0.368E-04	-792.8
8.0	0.177E-02	-510.6	0.262E-01	-345.3	0.429E-04	-797.3	0.351E-04	-793.4

Matrix Assembly Accessories:

File 'assemble.f'

```
C       THIS FILE CONTAINS SUBROUTINES FOR ASSEMBLING OF THE
C       SYSTEM MATRICES:
C       BBSORT, BISORT, IBSORT, DKIIMAT, KIIMAT, KNMDIAG
C       (replace sin/cos by trig. identities)


        SUBROUTINE BBSORT(MATFILE,XSYS,NAME,DIM,DIMJ,JDIM,ADD)
C*****************************************************************
C
C       PLACES THE MATRIX "*BB" (CALLED AS "NAME"),
C       READ FROM THE FILE "MATFILE",
C       ON THE MAIN DIAGONAL OF THE SYSTEM MATRIX "XSYS", PARTITION BB
C       (I.E. BLADE DYNAMIC SYSTEM, COEFFICIENTS OF THE BLADE DOF'S)
C       JDIM  -       NUMBER OF COLUMNS OF "NAME"
C       DIM   -       DIMENSION OF THE SYSTEM MATRIX "XSYS"
C       DIMJ  -       REDUCED COLUMN NUMBER; FOR CASES IN WHICH THE
C                     MATRICES ARE NOT SQUARE MATRICES, I.E. THE CONTROL
C                     MATRIX AND THE CONSTANT VECTOR-
C                     NO INFLOW COEFFICIENT PARTITION, BUT "BB"-
C                     AND "IB" TYPE ASSEMBLY;
C                     (DIM.NE.DIMJ) REQUIRES (ADD.EQ.1.AND.DIMJ.EQ.JDIM)
C       ADD   -       CONTROL SWITCH:
C                     0       MATRICES ON MAIN DIAGONAL
C                     1       MATRICES ONE UNDERNEATH THE OTHER ("ADDED")
C
C*****************************************************************
        INTEGER DIM,DIMJ, I, J,JDIM, Q, ADD, NADD, ISYS, JSYS
        REAL XSYS(DIM,DIMJ), DUM(JDIM)
        CHARACTER*4 NAME, DUMMY4
        CHARACTER*8 MATFILE, DUMMY8
C
        NADD=1-ADD
        IF(DIM.NE.DIMJ) THEN
          IF(NADD.EQ.0.AND.DIMJ.EQ.JDIM) GOTO 1
        ELSE
          IF(NADD.EQ.1) GOTO 1
        END IF
        PRINT*,'ERROR IN BBSORT:
     $ CHOICE OF "ADD"-OPTION AND MATRIX DIMENSIONS INCOMPATIBLE'
        PRINT*,'DIM,DIMJ,JDIM,ADD',DIM,DIMJ,JDIM,ADD
        STOP
 1      CONTINUE
C
```

```
      OPEN (UNIT=7,FILE=MATFILE)
      REWIND(7)
      READ(7,*) DUMMY8
      IF (DUMMY8.NE.MATFILE) THEN
        PRINT*,'ERROR WHILE ATTEMPTING TO READ MATFILE:'
        PRINT*,'FILENAME AND FILE CODE ARE NOT EQUIVALENT'
        STOP
      ELSE
      END IF
C
 5    CONTINUE
      READ(7,300)DUMMY4
      IF(DUMMY4.NE.NAME) GOTO 5
      DO 20 I=1,2
         READ(7,500)  (DUM(J),J=1,JDIM)
        DO 15 Q=1,4
              DO 10 J=1,JDIM
                 ISYS=2*(Q-1)+I
                 JSYS=JDIM*NADD*(Q-1)+J
                 XSYS(ISYS,JSYS)= DUM(J)
10            CONTINUE
15      CONTINUE
20    CONTINUE
      CLOSE(7)
300   FORMAT(A4)
500   FORMAT(10E14.6)
      RETURN
      END



      SUBROUTINE BISORT(MATFILE,XSYS,NAME,DIM,IDIM,DM,M0,KMAX,NCAP,
     $                  DIMINSM,DIMINM)
C*****************************************************************
C
C     PLACES THE MATRIX "*BIT" (CALLED AS "NAME"),
C     READ FROM THE FILE "MATFILE"(ATT.: STORED IN TRANSPOSED FORM!),
C     IN THE APPROPRIATE PARTITION OF THE SYSTEM MATRIX "XSYS",
C     (I.E. BLADE DYNAMIC SYSTEM, COEFFICIENTS OF THE INFLOW DOF'S)
C     THE ELEMENTS ARE MULTIPLIED BY COS(M*PSIQ) OR SIN(M*PSIQ),
C     RESPECTIVELY, AND ADDED TO THE SYSTEM MATRIX ACCORDING
C     TO HARMONIC NUMBER "M", POLYNOMIAL NUMBER "N", AND
C     BLADE NUMBER "Q"
C     THE ROUTINE ASSEMBLES TRUNCATED MATRICES AS A FUNCTION
C     OF THE MODEL BEING USED
C     IDIM   -      NUMBER OF ROWS OF "NAME"
C     DIM    -      DIMENSION OF THE SYSTEM MATRIX "XSYS"
C     DIMINM-       DIMENSION OF THE INFLOW PARTITION OF TRUNCATED
C                   SYSTEM (# OF INFLOW STATE VARIABLES)
C     DIMINSM-      DTO., SIN - PARTITION ONLY (BETAnm COEFFICIENTS)
C     NCAP   -      HIGHEST ORDER OF RADIAL POLYNOMIAL EXPANSION
C     M0     -      COEFFICIENTS FOR DETERMINATION OF COUPLING
```

```
C       DM    -       HARMONICS M FOR SPECIFIC MODEL:
C       KMAX-         M=M0+DM*K, K=0,1,...,KMAX
C
C*****************************************************************
        INTEGER I, J,IDIM, Q, M,DM,M0, KMAX,K,M1
        INTEGER COL,L, DIMINSM, DIMINM,NCAP,DIM
        REAL XSYS(DIM,DIM), PI, PSIQ,DUM(NCAP), TS, TC, SEARCHM
        CHARACTER*3 DUMMY3
        CHARACTER*4 NAME, DUMMY4
        CHARACTER*8 MATFILE, DUMMY8
        LOGICAL EVENQ, EVENM1
C
        OPEN (UNIT=7,FILE=MATFILE)
        REWIND(7)
        READ(7,*) DUMMY8
        IF (DUMMY8.NE.MATFILE) THEN
          PRINT*,'ERROR WHILE ATTEMPTING TO READ MATFILE:'
          PRINT*,'FILENAME AND FILE CODE ARE NOT EQUIVALENT'
          STOP
        ELSE
        END IF
C
 5      CONTINUE
        READ(7,300)DUMMY4
        IF(DUMMY4.NE.NAME) GOTO 5
        PRINT*,'              BISORT:  found  ',dummy4
        READ(7,300)DUMMY4
        COL=8
        DO 35 K=0,KMAX
          M=DM*K+M0
           M1=M+1-4*(M/4)
 10       CONTINUE
          READ(7,500)DUMMY3,SEARCHM
          SEARCHM=SEARCHM*10**6
          IF((DUMMY3.NE.'MI=').OR.(SEARCHM.NE.REAL(M))) GOTO 10
          JMAX=(NCAP-M)/2
         DO 30 J=0,JMAX
              READ(7,510)NJ,(DUM(I),I=1,IDIM)
              DO 25 Q=1,4
                    EVENQ=( (REAL(Q)/2.) .EQ. REAL(Q/2) )
                    EVENM1=( (REAL(M1)/2.) .EQ. REAL(M1/2) )
                    IF(M1.EQ.1.OR.Q.EQ.1.OR.(M1.EQ.3.AND.Q.EQ.3)) THEN
                      TC=1.
                    ELSE IF ( ( (.NOT.EVENM1).AND.EVENQ )
     $                   .OR.( (.NOT.EVENQ).AND.EVENM1 )  ) THEN
                      TC=-1.
                    ELSE
                      TC=0.
                    END IF
                    IF(EVENM1.AND.EVENQ) THEN
                      IF(M1.EQ.Q) THEN
                          TS=1.
```

```
                    ELSE
                        TS=-1.
                    END IF
                  ELSE
                    TS=0.
                  END IF
C
                  IF(M.EQ.0) THEN
                    DO 15 L=1,IDIM
                  XSYS(IDIM*(Q-1)+L,COL+J+1)=TC*DUM(L)
 15                 CONTINUE
                    GOTO 25
                  ELSE
                    DO 20 L=1,IDIM
                  XSYS(IDIM*(Q-1)+L,COL+J+1)=TC*DUM(L)
                        XSYS(IDIM*(Q-1)+L,COL+DIMINSM+J+1)=TS*DUM(L)
 20                 CONTINUE
                  END IF
 25           CONTINUE
 30       CONTINUE
          COL=COL+JMAX+1
 35       CONTINUE
          CLOSE(7)
 300      FORMAT(A4)
 500      FORMAT(A3,E14.6)
 510      FORMAT(I3,3E14.6)
          RETURN
          END



      SUBROUTINE IBSORT(MATFILE,XSYS,NAME,DIM,DIMJ,JDIM,DM,
     $                  M0,KMAX,NCAP,DIMINSM,DIMINM,ADD)
C*****************************************************************
C
C     PLACES THE MATRIX "*IBT" (CALLED AS "NAME"),
C     READ FROM THE FILE "MATFILE",
C     IN THE APPROPRIATE PARTITION OF THE SYSTEM MATRIX "XSYS",
C     (I.E. INFLOW DYNAMIC SYSTEM, COEFFICIENTS OF THE BLADE DOF'S)
C     THE ELEMENTS ARE MULTIPLIED BY COS(M*PSIQ) OR SIN(M*PSIQ),
C     RESPECTIVELY, AND ADDED TO THE SYSTEM MATRIX ACCORDING
C     TO HARMONIC NUMBER "M", POLYNOMIAL NUMBER "N", AND
C     BLADE NUMBER "Q"
C     THE ROUTINE ASSEMBLES TRUNCATED MATRICES AS A FUNCTION
C     OF THE MODEL BEING USED
C     JDIM  -       NUMBER OF COLUMNS OF "NAME"
C     DIM   -       DIMENSION OF THE SYSTEM MATRIX "XSYS"
C     DIMJ  -       REDUCED COLUMN NUMBER; FOR CASES IN WHICH THE
C                   MATRICES ARE NOT SQUARE MATRICES, I.E. THE CONTROL
```

```
C                   MATRIX AND THE CONSTANT VECTOR-
C                   NO INFLOW COEFFICIENT PARTITION, BUT "BB"-
C                   AND "IB" TYPE ASSEMBLY;
C                   (DIM.NE.DIMJ) REQUIRES (ADD.EQ.0.AND.DIMJ.EQ.JDIM)
C       DIMINM-     DIMENSION OF THE INFLOW PARTITION OF TRUNCATED
C                   SYSTEM (# OF INFLOW STATE VARIABLES)
C       DIMINSM-    DTO., SIN - PARTITION ONLY (BETAnm COEFFICIENTS)
C       NCAP  -     HIGHEST ORDER OF RADIAL POLYNOMIAL EXPANSION
C       M0    -     COEFFICIENTS FOR DETERMINATION OF COUPLING
C       DM    -      HARMONICS M FOR SPECIFIC MODEL:
C       KMAX-       M=M0+DM*K, K=0,1,...,KMAX
C       ADD   -     CONTROL SWITCH:
C                   0       MATRICES SORTED ACCORDING TO BLADE NUMBER
C                   1       MATRICES ADDED
C
C*****************************************************************
      INTEGER I, J,JDIM, Q, M,M1,DM,M0, KMAX,K
      INTEGER ROW,L, DIMINSM, DIMINM,NCAP,DIM,DIMJ,NI, NADD,ADD
      REAL XSYS(DIM,DIMJ), PI, PSIQ,DUM(NCAP), TS, TC, SEARCHM
      CHARACTER*3 DUMMY3
      CHARACTER*4 NAME, DUMMY4
      CHARACTER*8 MATFILE, DUMMY8
      LOGICAL EVENM1, EVENQ
C
      NADD=1-ADD
      IF(DIM.NE.DIMJ) THEN
        IF(NADD.EQ.0.AND.DIMJ.EQ.JDIM) GOTO 1
      ELSE
        IF(NADD.EQ.1) GOTO 1
      END IF
      PRINT*,'ERROR IN IBSORT:
     $ CHOICE OF "ADD"-OPTION AND MATRIX DIMENSIONS INCOMPATIBLE'
      PRINT*,'DIM,DIMJ,JDIM,ADD',DIM,DIMJ,JDIM,ADD
      STOP
 1    CONTINUE
C
      OPEN (UNIT=7,FILE=MATFILE)
      REWIND(7)
      READ(7,*) DUMMY8
      IF (DUMMY8.NE.MATFILE) THEN
        PRINT*,'ERROR WHILE ATTEMPTING TO READ MATFILE:'
        PRINT*,'FILENAME AND FILE CODE ARE NOT EQUIVALENT'
        STOP
      ELSE
      END IF
C
 5    CONTINUE
      READ(7,300)DUMMY4
      IF(DUMMY4.NE.NAME) GOTO 5
      PRINT*,'              IBSORT:  found  ',dummy4
      READ(7,300)DUMMY4
      ROW=8
```

```
      DO 35 K=0,KMAX
        M=DM*K+M0
         M1=M+1-4*(M/4)
10      CONTINUE
        READ(7,500)DUMMY3,SEARCHM
        SEARCHM=SEARCHM*10**6
        IF((DUMMY3.NE.'MI=').OR.(SEARCHM.NE.REAL(M))) GOTO 10
        IMAX=(NCAP-M)/2
        DO 30 I=0,IMAX
            READ(7,510)NI,(DUM(J),J=1,JDIM)
            DO 25 Q=1,4
                EVENQ=( (REAL(Q)/2.) .EQ. REAL(Q/2) )
                EVENM1=( (REAL(M1)/2.) .EQ. REAL(M1/2) )
                IF(M1.EQ.1.OR.Q.EQ.1.OR.(M1.EQ.3.AND.Q.EQ.3)) THEN
                  TC=1.
                ELSE IF ( ( (.NOT.EVENM1).AND.EVENQ )
     $              .OR.( (.NOT.EVENQ).AND.EVENM1 ) ) THEN
                  TC=-1.
                ELSE
                  TC=0.
                END IF
                IF(EVENM1.AND.EVENQ) THEN
                  IF(M1.EQ.Q) THEN
                      TS=1.
                  ELSE
                      TS=-1.
                  END IF
                ELSE
                  TS=0.
                END IF
                IF(M.EQ.0) THEN
                  DO 15 L=1,JDIM
                      XSYS(ROW+I+1,NADD*JDIM*(Q-1)+L)=
     $                 XSYS(ROW+I+1,NADD*JDIM*(Q-1)+L)+0.5*DUM(L)
15                  CONTINUE
                  GOTO 25
                ELSE
                  DO 20 L=1,JDIM
                      XSYS(ROW+I+1,NADD*JDIM*(Q-1)+L)=
     $                 XSYS(ROW+I+1,NADD*JDIM*(Q-1)+L)+TC*DUM(L)
                      XSYS(ROW+DIMINSM+I+1,NADD*JDIM*(Q-1)+L)=
     $                 XSYS(ROW+DIMINSM+I+1,NADD*JDIM*(Q-1)+L)
     $                                                  +TS*DUM(L)
20                  CONTINUE
                END IF
25          CONTINUE
30      CONTINUE
        ROW=ROW+IMAX+1
35    CONTINUE
      CLOSE(7)
300   FORMAT(A4)
500   FORMAT(A3,E14.6)
```

```
510   FORMAT(I3,3E14.6)
      RETURN
      END




      SUBROUTINE DKIIMAT(MATFILE,KSYS,DIM,DM,M0,KMAX,NCAP,
     $                   DIMINSM,DIMINM)
C*****************************************************************
C
C     ADDS THE MATRIX "DKII",
C     READ FROM THE FILE "MATFILE",
C     TO THE APPROPRIATE PARTITIONS OF THE SYSTEM MATRIX "KSYS",
C     (I.E. INFLOW DYNAMIC SYSTEM, COEFFICIENTS OF THE INFLOW DOF'S)
C     THE ELEMENTS ARE MULTIPLIED BY COS(MI*PSIQ)* COS(MJ*PSIQ), ...
C     SIN(MI*PSIQ)*SIN(MJ*PSIQ); APPLYING TRIGONOMETRIC IDENTITIES
C     FOR THE CASE FOUR BLADES) ACCORDING TO VERTICAL PARTITION "MI"
C     AND HORIZONTAL PARTITION "MJ", AND ALL BLADE CONTRIBUTIONS ARE
C     ADDED UP
C     THE ROUTINE ASSEMBLES TRUNCATED MATRICES AS A FUNCTION
C     OF THE MODEL BEING USED
C
C     VERSION NOVEMBER 29, 1990: READS ONLY UPPER
C                                TRIANGULAR PARTITIONS
C                                OF DKII FROM MATFILE, ASSEMBLES THEN
C                                COMPLETE DKII (USE SYMMETRY)
C
C     DIM   -      DIMENSION OF THE SYSTEM MATRIX "KSYS"
C     DIMINM-      DIMENSION OF THE INFLOW PARTITION OF TRUNCATED
C                  SYSTEM (# OF INFLOW STATE VARIABLES)
C     DIMINSM-     DTO., SIN - PARTITION ONLY (BETAnm COEFFICIENTS)
C     KMAX-        COUNTER FOR MATRIX PARTITIONS
C     NCAP  -      HIGHEST ORDER OF RADIAL POLYNOMIAL EXPANSION
C     M0    -      COEFFICIENTS FOE DETERMINATION OF COUPLING
C     DM    -      HARMONICS M FOR SPECIFIC MODEL:
C     KMAX-        M=M0+DM*K, K=0,1,...,KMAX
C
C*****************************************************************
      INTEGER I, J,IMAX,JMAX,MI,MJ,DM,M0,NI,KMAX,K,R,Q
      INTEGER ROW,COL,DIMINSM, DIMINM,NCAP,DIM
      REAL KSYS(DIM,DIM), DKSYS(DIMINM,DIMINM)
      REAL DUM(NCAP),TC,T11,T12,T22
      REAL SEARCHMI,SEARCHMJ
      CHARACTER*3 DUMMY3I,DUMMY3J
      CHARACTER*4 DUMMY4
      CHARACTER*8 MATFILE, DUMMY8
      LOGICAL EVENMI, EVENMJ,DMIM4
C
```

```
      OPEN (UNIT=7,FILE=MATFILE)
      REWIND(7)
      READ(7,*) DUMMY8
      IF (DUMMY8.NE.MATFILE) THEN
        PRINT*,'ERROR WHILE ATTEMPTING TO READ MATFILE:'
        PRINT*,'FILENAME AND FILE CODE ARE NOT EQUIVALENT'
        STOP
      ELSE
      END IF
C
      DO 2 I=1,DIMINM
        DO 1 J=1,DIMINM
            DKSYS(I,J)=0.
1       CONTINUE
2     CONTINUE
C
5     CONTINUE
      READ(7,300)DUMMY4
      IF(DUMMY4.NE.'DKII') GOTO 5
      PRINT*,'                DKIIMAT:  found  ',dummy4
      ROW=0
      DO 35 K=0,KMAX
        MI=DM*K+M0
        IMAX=(NCAP-MI)/2
        COL=ROW
C
        DO 30 R=K,KMAX
            MJ=DM*R+M0
            JMAX=(NCAP-MJ)/2
            DMIM4=(  (REAL(MI-MJ)/4.)  .EQ.  REAL((MI-MJ)/4)  )
            EVENMI=((REAL(MI)/2.).EQ.REAL(MI/2))
            EVENMJ=((REAL(MJ)/2.).EQ.REAL(MJ/2))
            IF(MI.EQ.0) THEN
              TC=0.5
            ELSE
              TC=1.
           END IF
            IF(EVENMI.AND.(DMIM4.OR.(MI.EQ.MJ))) THEN
              T11=TC*4.
            ELSE IF ( (.NOT.(EVENMI)) .AND. (.NOT.(EVENMJ)) ) THEN
              T11=TC*2.
            ELSE
              T11=0.
            END IF
            T12=0.
            IF ( (.NOT.(EVENMI)) .AND. (.NOT.(EVENMJ)) ) THEN
              IF((MI.EQ.MJ).OR.DMIM4) THEN
                  T22=2.
             ELSE
                  T22=-2.
             END IF
            ELSE
```

```
                T22=0.
              END IF
C
 10           CONTINUE
              READ(7,500)DUMMY3I,SEARCHMI
              SEARCHMI=SEARCHMI*10**6
              IF((DUMMY3I.NE.'MI=').OR.(SEARCHMI.NE.REAL(MI))) THEN
                GOTO 10
              ELSE
                READ(7,500)DUMMY3J,SEARCHMJ
                SEARCHMJ=SEARCHMJ*10**6
                IF((DUMMY3J.NE.'MJ=').OR.(SEARCHMJ.NE.REAL(MJ))) GOTO 10
              END IF
C
            DO 25 I=0,IMAX
                READ(7,510)NI,(DUM(J),J=0,JMAX)
              DO 20 J=0,JMAX
                IF(MI.EQ.0.OR.MJ.EQ.0) THEN
                    DKSYS(ROW+1+I,COL+1+J)=T11*DUM(J)
                ELSE
                    DKSYS(ROW+1+I,COL+1+J)=T11*DUM(J)
                    DKSYS(ROW+1+I,COL+DIMINSM+1+J)=T12*DUM(J)
                    DKSYS(ROW+DIMINSM+1+I,COL+DIMINSM+1+J)
   $                                                    =T22*DUM(J)
                END IF
 20           CONTINUE
 25         CONTINUE
          COL=COL+JMAX+1
 30       CONTINUE
          ROW=ROW+IMAX+1
 35     CONTINUE
        CLOSE(7)
C
        DO 45 I=1,DIMINM-1
          DO 40 J=I+1,DIMINM
              DKSYS(J,I)=DKSYS(I,J)
 40       CONTINUE
 45     CONTINUE
        DO 55 I=1,DIMINM
          DO 50 J=1,DIMINM
              KSYS(I+8,J+8)=KSYS(I+8,J+8)+DKSYS(I,J)
 50       CONTINUE
 55     CONTINUE
C
 300    FORMAT(A4)
 500    FORMAT(A3,E14.6)
 510    FORMAT(I3,20E14.6)
        RETURN
        END
```

```
      SUBROUTINE KIIMAT(MATFILE,KSYS,DIM,DM,M0,KMAX,NCAP,
     $                  DIMINSM,DIMINM,V,VT)
C*****************************************************************
C
C     THE ROUTINE ASSEMBLES TRUNCATED KII MATRICES AS A FUNCTION
C     OF THE MODEL "MODEL" BEING USED, READING THE APPARENT MASS
C     DIAGONALS Kmn AND MATRICES Bmnj FROM THE FILE "MATFILE"
C     DIM   -     DIMENSION OF THE SYSTEM MATRIX "KSYS"
C     DIMINM-     DIMENSION OF THE INFLOW PARTITION OF TRUNCATED
C                 SYSTEM (# OF INFLOW STATE VARIABLES)
C     DIMINSM     -       DTO., SIN - PARTITION ONLY (BETAnm COEFFICIENTS)
C     M0    -     COEFFICIENTS FOE DETERMINATION OF COUPLING
C     DM    -     HARMONICS M FOR SPECIFIC MODEL:
C     KMAX        -       M=M0+DM*K, K=0,1,...,KMAX
C
C*****************************************************************
      INTEGER I, J,JMAX,M,DM,M0, KMAX,K,DIM
      INTEGER ROWUP,ROWDW,COLUP,COLDW
      INTEGER DIMINSM, DIMINM,NCAP,NJ,LIN
      REAL KSYS(DIM,DIM), DUM(NCAP), SEARCHM, V,VT,VC
      CHARACTER*3 DUMMY3
      CHARACTER*4 DUMMY4
      CHARACTER*8 MATFILE, DUMMY8
C
      OPEN (UNIT=7,FILE=MATFILE)
      REWIND(7)
      READ(7,*) DUMMY8
      IF (DUMMY8.NE.MATFILE) THEN
        PRINT*,'ERROR WHILE ATTEMPTING TO READ MATFILE:'
        PRINT*,'FILENAME AND FILE CODE ARE NOT EQUIVALENT'
        STOP
      ELSE
      END IF
C
C     KNM
C
 5    CONTINUE
      READ(7,300)DUMMY4
      IF(DUMMY4.NE.'KNM ') GOTO 5
      PRINT*,'              KIIMAT:   found ',dummy4
      READ(7,300)DUMMY4
      ROWUP=8
      ROWDW=ROWUP+DIMINSM
      COLUP=ROWDW
      COLDW=ROWUP
      DO 20 K=0,KMAX
        M=DM*K+M0
        IF(M.EQ.0) THEN
```

```
            ROWUP=ROWUP+NCAP/2+1
            ROWDW=ROWDW+NCAP/2+1
            COLUP=COLUP+NCAP/2+1
            COLDW=COLDW+NCAP/2+1
          ELSE
 10         CONTINUE
            READ(7,500)DUMMY3,SEARCHM
            SEARCHM=SEARCHM*10**6
            IF((DUMMY3.NE.'MI=').OR.(SEARCHM.NE.REAL(M))) GOTO 10
            JMAX=(NCAP-M)/2
            DO 15 J=0,JMAX
               READ(7,500)NJ,DUM(1)
              KSYS(ROWUP+J+1,COLUP+J+1)=
     $          KSYS(ROWUP+J+1,COLUP+J+1)-M*DUM(1)
             KSYS(ROWDW+J+1,COLDW+J+1)=
     $          KSYS(ROWDW+J+1,COLDW+J+1)+M*DUM(1)
 15         CONTINUE
            ROWUP=ROWUP+JMAX+1
            ROWDW=ROWDW+JMAX+1
            COLUP=COLUP+JMAX+1
            COLDW=COLDW+JMAX+1
          END IF
 20     CONTINUE
C
C       BMNJ
C
 25     CONTINUE
        READ(7,300)DUMMY4
        IF(DUMMY4.NE.'BMNJ') GOTO 25
        PRINT*,'              KIIMAT:   found  ',dummy4
        READ(7,300)DUMMY4
        ROW=8
        COL=ROW
        DO 45 K=0,KMAX
          M=DM*K+M0
 30       CONTINUE
          READ(7,500)DUMMY3,SEARCHM
          SEARCHM=SEARCHM*10**6
          IF((DUMMY3.NE.'MI=').OR.(SEARCHM.NE.REAL(M))) GOTO 30
          JMAX=(NCAP-M)/2
          DO 40 I=0,JMAX
            READ(7,510)NJ,(DUM(J),J=0,JMAX)
            IF(NJ.EQ.1)THEN
              VC=VT
            ELSE
              VC=V
            END IF
            DO 35 J=0,JMAX
              IF(M.EQ.0) THEN
                  KSYS(ROW+I+1,COL+J+1) =
     $            KSYS(ROW+I+1,COL+J+1)+VC*DUM(J)
              ELSE
```

```
                KSYS(ROW+I+1,COL+J+1) =
     $          KSYS(ROW+I+1,COL+J+1)+ V*DUM(J)
                KSYS(ROW+DIMINSM+I+1,COL+DIMINSM+J+1) =
     $          KSYS(ROW+DIMINSM+I+1,COL+DIMINSM+J+1) + V*DUM(J)
             END IF
35         CONTINUE
40       CONTINUE
         ROW=ROW+JMAX+1
         COL=COL+JMAX+1
45     CONTINUE
       CLOSE(7)
C
300    FORMAT(A4)
500    FORMAT(A3,E14.6)
510    FORMAT(I3,20E14.6)
       RETURN
       END




       SUBROUTINE KNMDIAG(MATFILE,CSYS,DIM,DM,M0,KMAX,
     $                          NCAP,DIMINSM,DIMINM)
C*****************************************************************
C
C      THE ROUTINE ASSEMBLES TRUNCATED APPARENT MASS
C      DIAGONAL MATRIX OF THE MODEL BEING USED, READING THE APPARENT
C      MASS DIAGONALS Kmn FROM THE FILE "MATFILE"
C      DIM    -       DIMENSION OF THE SYSTEM MATRIX "CSYS"
C      DIMINM-        DIMENSION OF THE INFLOW PARTITION OF TRUNCATED
C                     SYSTEM (# OF INFLOW STATE VARIABLES)
C      DIMINSM-       DTO., SIN - PARTITION ONLY (BETAnm COEFFICIENTS)
C      M0     -       COEFFICIENTS FOR DETERMINATION OF COUPLING
C      DM     -       HARMONICS M FOR SPECIFIC MODEL:
C      KMAX-          M=M0+DM*K, K=0,1,...,KMAX
C
C*****************************************************************
       INTEGER J,JMAX,M,DM,M0, KMAX,K
       INTEGER ROW,COL,DIM,DIMINSM, DIMINM,NCAP,NJ
       REAL CSYS(DIM,DIM), DUM, SEARCHM
       CHARACTER*3 DUMMY3
       CHARACTER*4 DUMMY4
       CHARACTER*8 MATFILE, DUMMY8
C
       OPEN (UNIT=7,FILE=MATFILE)
       REWIND(7)
       READ(7,*) DUMMY8
       IF (DUMMY8.NE.MATFILE) THEN
```

```
        PRINT*,'ERROR WHILE ATTEMPTING TO READ MATFILE:'
        PRINT*,'FILENAME AND FILE CODE ARE NOT EQUIVALENT'
        STOP
      ELSE
      END IF
5     CONTINUE
      READ(7,300)DUMMY4
      IF(DUMMY4.NE.'KNM ') GOTO 5
      PRINT*,'              KNMDIAG:  found  ',dummy4
      READ(7,300)DUMMY4
      ROW=8
      COL=ROW
      DO 20 K=0,KMAX
        M=DM*K+M0
10      CONTINUE
        READ(7,500)DUMMY3,SEARCHM
        SEARCHM=SEARCHM*10**6
        IF((DUMMY3.NE.'MI=').OR.(SEARCHM.NE.REAL(M))) GOTO 10
        JMAX=(NCAP-M)/2
        DO 15 J=0,JMAX
            READ(7,510)NJ,DUM
            IF(M.EQ.0) THEN
              CSYS(ROW+J+1,COL+J+1)=DUM
            ELSE
              CSYS(ROW+J+1,COL+J+1)=DUM
              CSYS(ROW+DIMINSM+J+1,COL+DIMINSM+J+1)=DUM
            END IF
15      CONTINUE
        ROW=ROW+JMAX+1
        COL=COL+JMAX+1
20    CONTINUE
      CLOSE(7)
C
300   FORMAT(A3)
500   FORMAT(A3,E14.6)
510   FORMAT(I3,E14.6)
      RETURN
      END
```

Table 1: Comparison of Eigenvalues

Frequency (per rev)	6F [13]	Myklestad [13]	System I	System II	System III
Lagging	0.209	0.206	-	-	-
Flapping	1.022	1.021	1.0224	1.0224	1.0224
Feathering	1.254	1.480	1.2578	1.2590	1.2591
Bending	3.180	3.397	-	-	-
Twisting	13.41	13.93	-	-	-
Servo Flap	15.27	-	-	14.80	-

Table 2: Roots of Transfer Functions without Inflow Dynamics

	System I	System II	System III
Flapping			
quasi-steady			
- frequency	1.1675	0.9668	0.9623
- damping	0.4250	0.3069	0.3280
incl. app. mass			
- frequency	1.2139	0.9888	0.9894
- damping	0.5632	0.3206	0.3475
Feathering			
quasi-steady			
- frequency	1.1452	1.4981	1.4755
- damping	0.1003	0.2005	0.1880
incl. app. mass			
- frequency	1.0916	1.4512	1.4222
- damping	0.1263	0.3207	0.3011
Flap Defl.			
quasi-steady			
- frequency	-	14.6892	-
- damping	-	-0.00008	-
incl. app. mass			
- frequency	-	14.6817	-
- damping	-	0.0074	-

Table 3: Number of Inflow State Variables for a Given Power of $\bar{r}$

Highest Power of $\bar{r}$	m											Total Number of Inflow States
	0	1	2	3	4	5	6	7	8	9	10	
0	1											1
1	1	1										3
2	2	1	1									6
3	2	2	1	1								10
4	3	2	2	1	1							15
5	3	3	2	2	1	1						21
6	4	3	3	2	2	1	1					28
7	4	4	3	3	2	2	1	1				36
8	5	4	4	3	3	2	2	1	1			45
9	6	5	4	4	3	3	2	2	1	1		55
10	7	6	5	4	4	3	3	2	2	1	1	66

Table 4: Convergence of Trim States

State Variable	N = 0	N = 4	N = 8	N = 12
δ_c [rad]	-0.01190	-0.023280	-0.030826	-0.035849
β [rad]	0.045406	0.044389	0.043839	0.043971
θ [rad]	0.12244	0.13119	0.13724	0.14158
$a^m{}_n$; m, n =				
0, 1	*0.034641*	*0.034641*	*0.034641*	*0.034641*
0, 11	-	-	-	0.000305
4, 5	-	0.003390	0.002922	0.002828
4, 9	-	-	0.002911	0.002011
4, 11	-	-	-	0.001247
4, 13	-	-	-	0.001266
8, 9	-	-	0.001818	0.001542
8, 13	-	-	-	0.001554
12, 13	-	-	-	0.001276

(all other inflow states equal to zero or less then 10^{-8})

Table 5: Frequency Response Comparison in Flapping

N = 8

Amplitude[10^{-2}] (% of 'No Inflow'-Amp. for same ω_R)	ω_R = 3/rev	ω_R = 4/rev	ω_R = 5/rev
No Inflow Dynamics	3.57 (100)	1.30 (100)	0.647 (100)
collective	2.70 (76)	**0.782** (**61**)	0.468 (72)
progressing	**1.98** (**55**)	0.912 (70)	0.521 (81)
regressing	2.90 (81)	0.927 (71)	**0.367** (**57**)
differential,	2.62 (73)	1.04 (80)	0.457 (70)

Table 6: Frequency Response Comparison in Flapping

N = 12

Amplitude[10^{-2}] (% of 'No Inflow'-Amp. for same ω_R)	ω_R = 3/rev	ω_R = 4/rev	ω_R = 5/rev
No Inflow Dynamics	3.62 (100)	1.32 (100)	0.657 (100)
collective	2.65 (74)	**0.78** (**60**)	0.458 (71)
progressing	**1.94** (**55**)	0.891 (69)	0.480 (75)
regressing	2.83 (80)	0.885 (69)	**0.342** (**53**)
differential	2.57 (72)	1.00 (78)	0.424 (66)

Table 7: Frequency Response Comparison in Feathering

N = 12

Amplitude[1]	ω_R = 3/rev	ω_R = 4/rev	ω_R = 5/rev
No Inflow Dyn.	0.260	0.132	0.0792
collective	0.262	0.133	0.0796
progressing	0.262	0.133	0.0794
regressing	0.262	0.133	0.0796
differential	0.262	0.132	0.0793

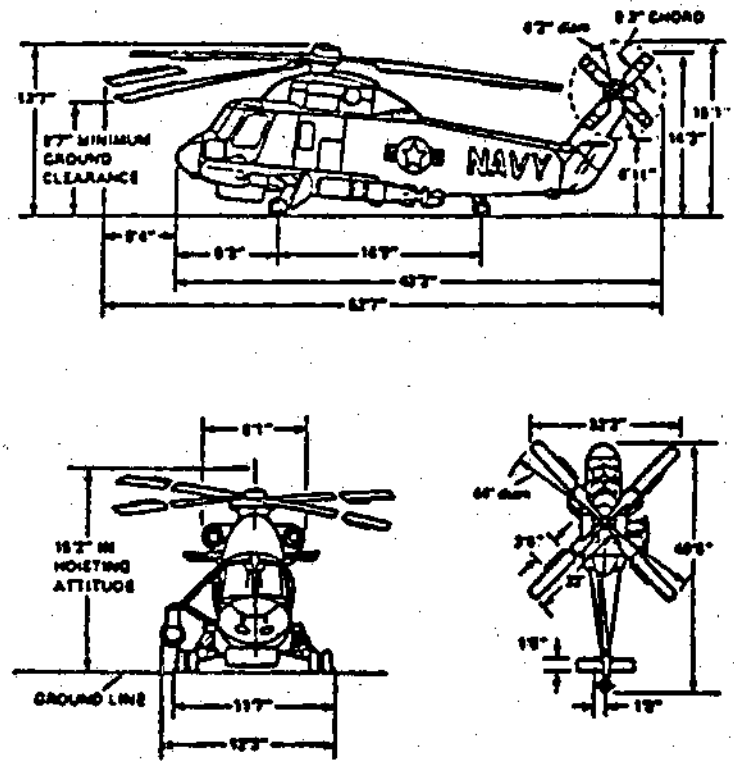


Fig. 1: Kaman SH-2F Helicopter [3]

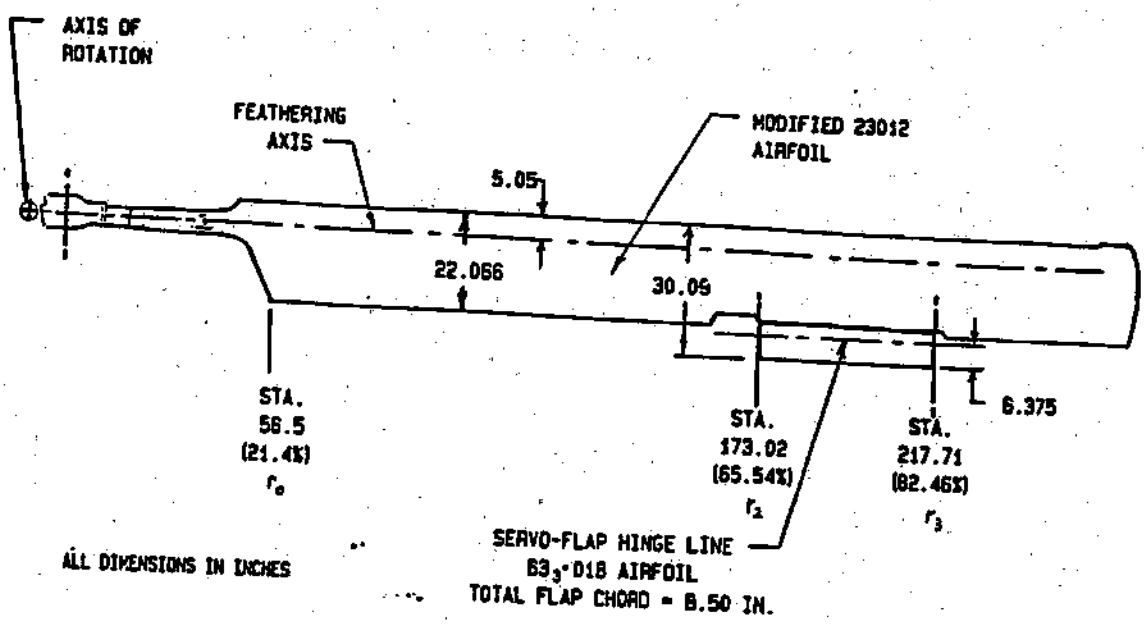


Fig. 2: SH-2F / 101 Rotor Blade Configuration [14]

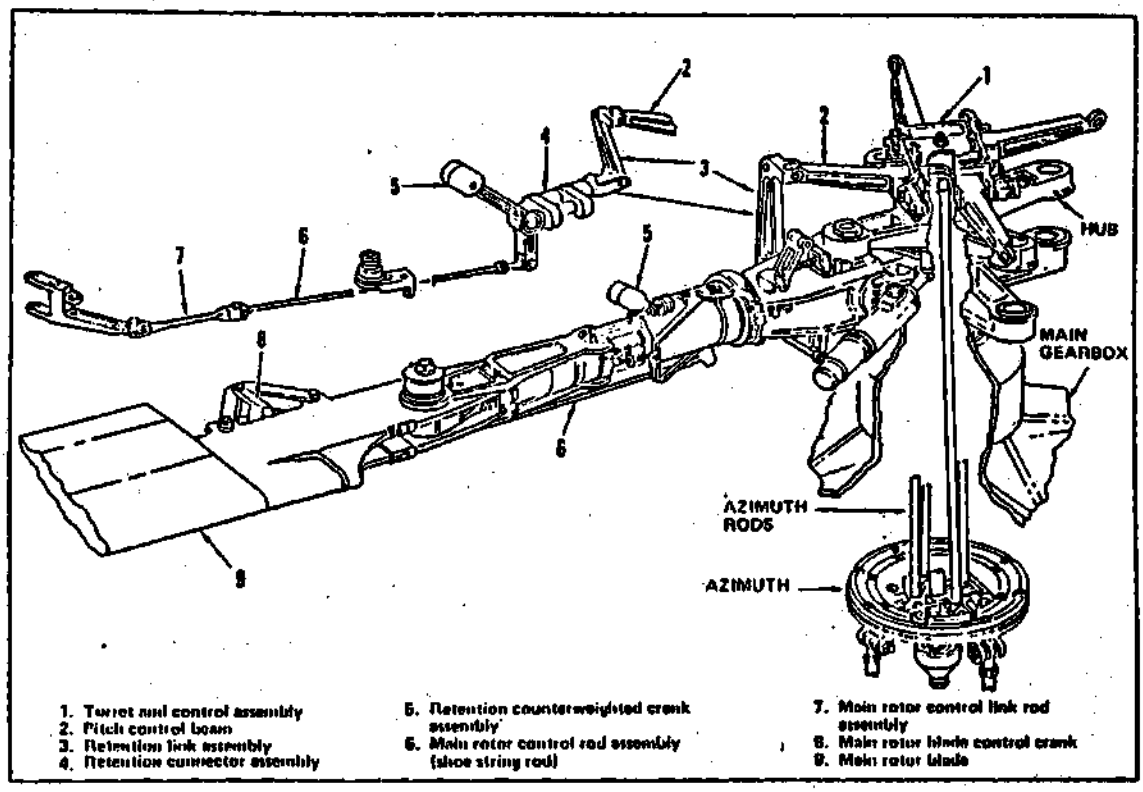


Fig. 3: 101 Rotor Control System

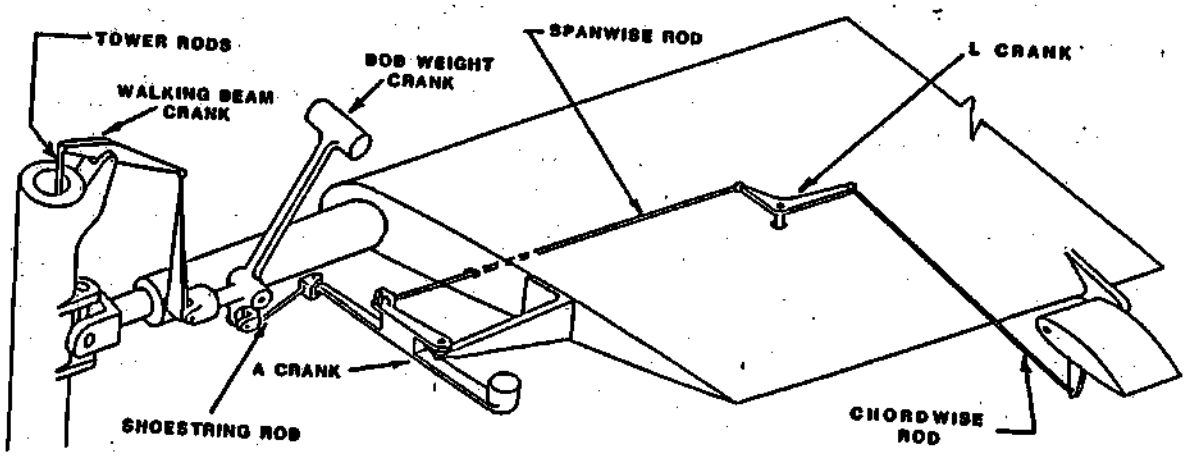


Fig. 4: CMRB Control Linkage Schematic

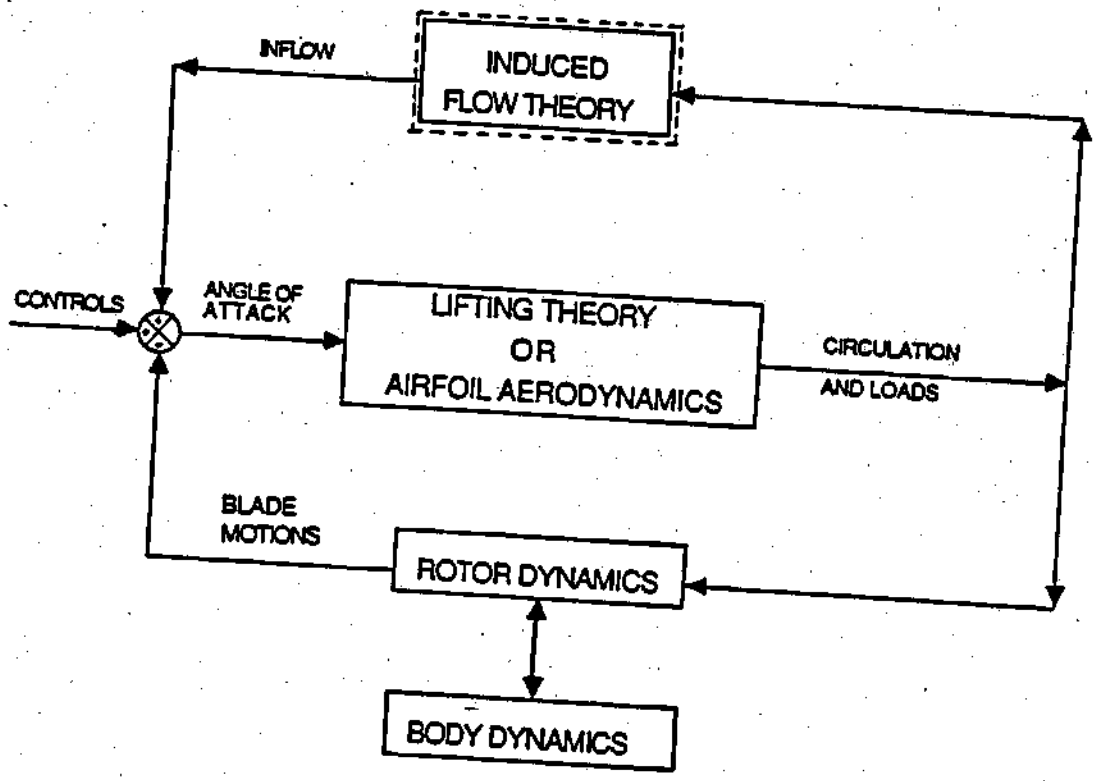


Fig. 5: **Block Diagram of Inflow Dynamics as Open Loop and Closed Loop Systems [12]**

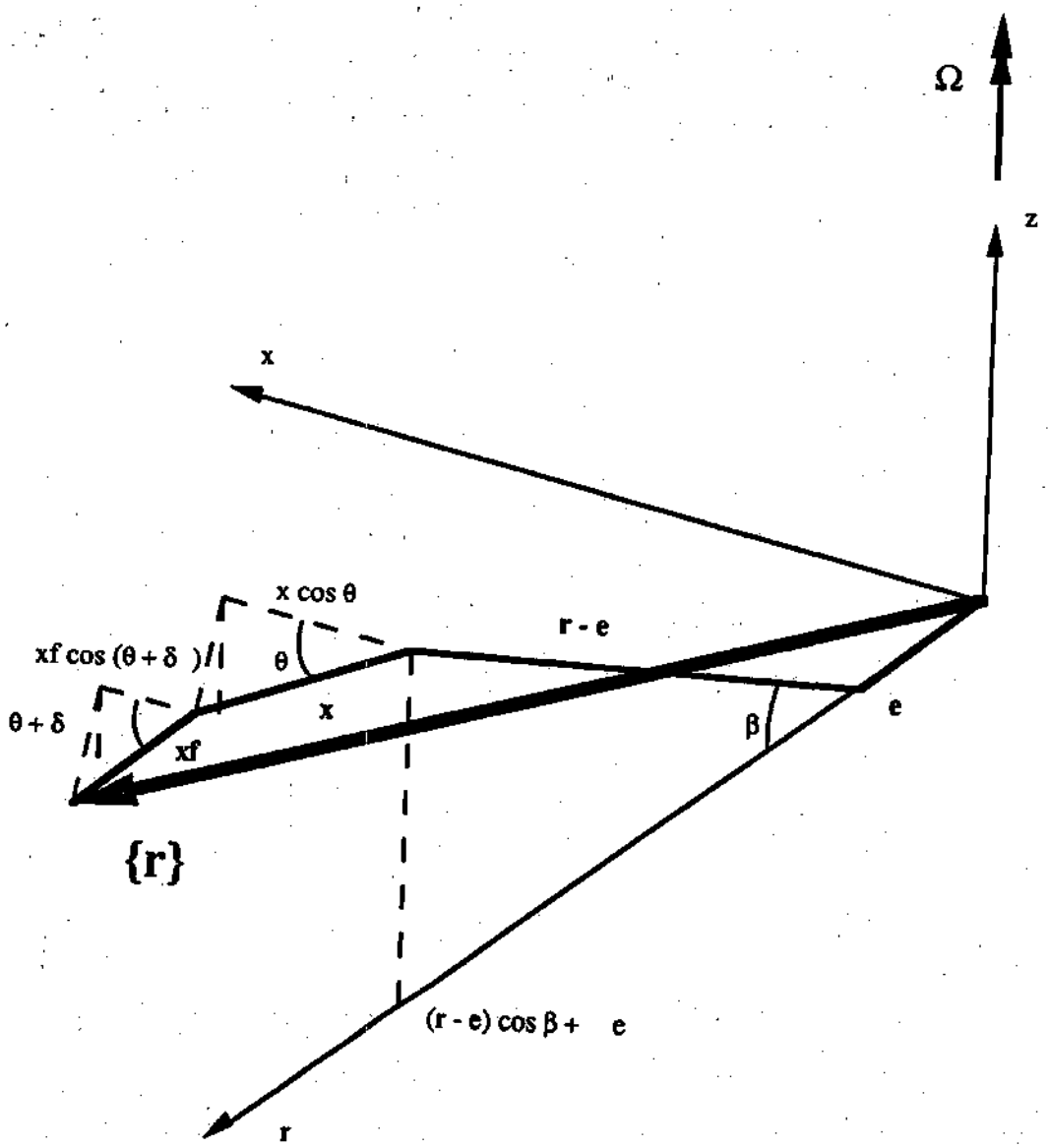


Fig. 6: Sketch of Blade / Servo Flap Kinematics

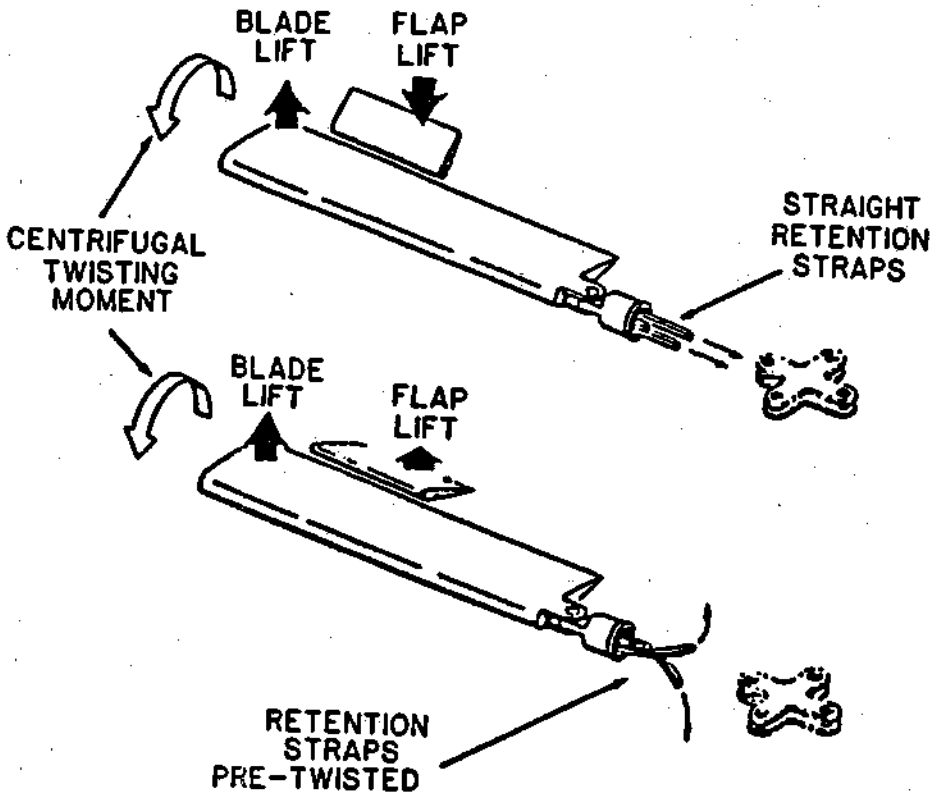


Fig. 7: Retention Strap Arrangement

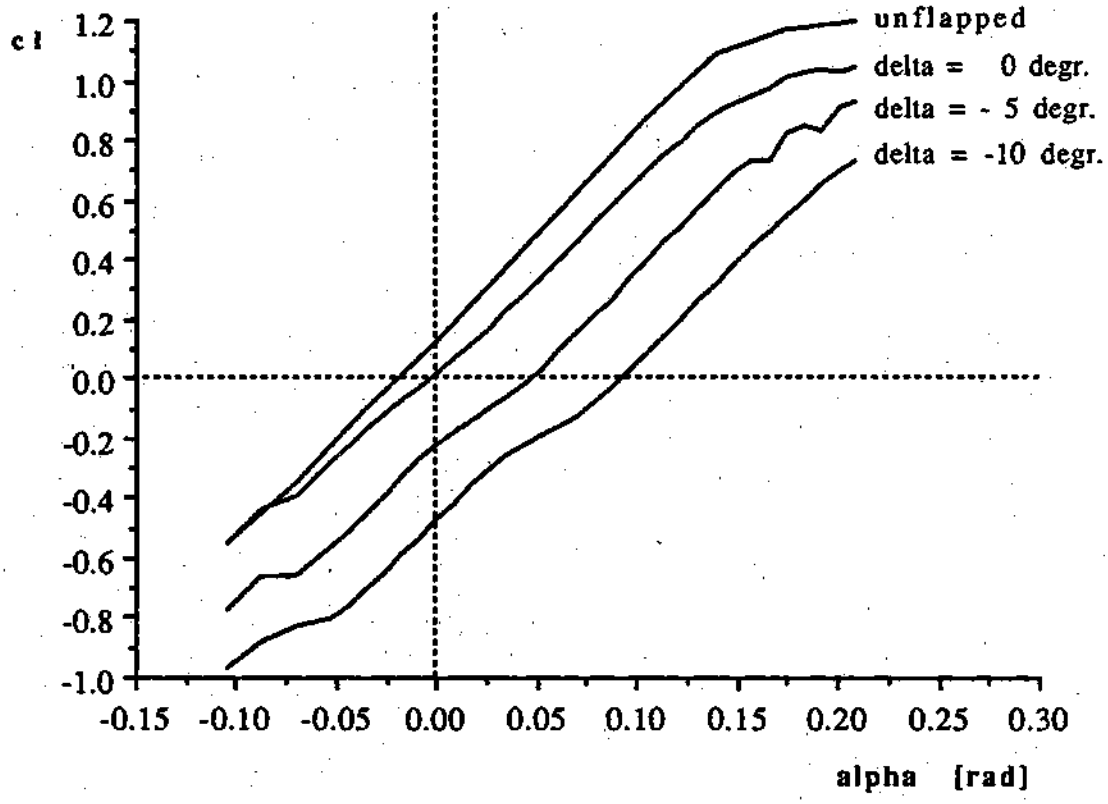


Fig. 8: Kaman 101 Rotor Blade and Blade/Flap

Lift Coefficient, M = 0.5

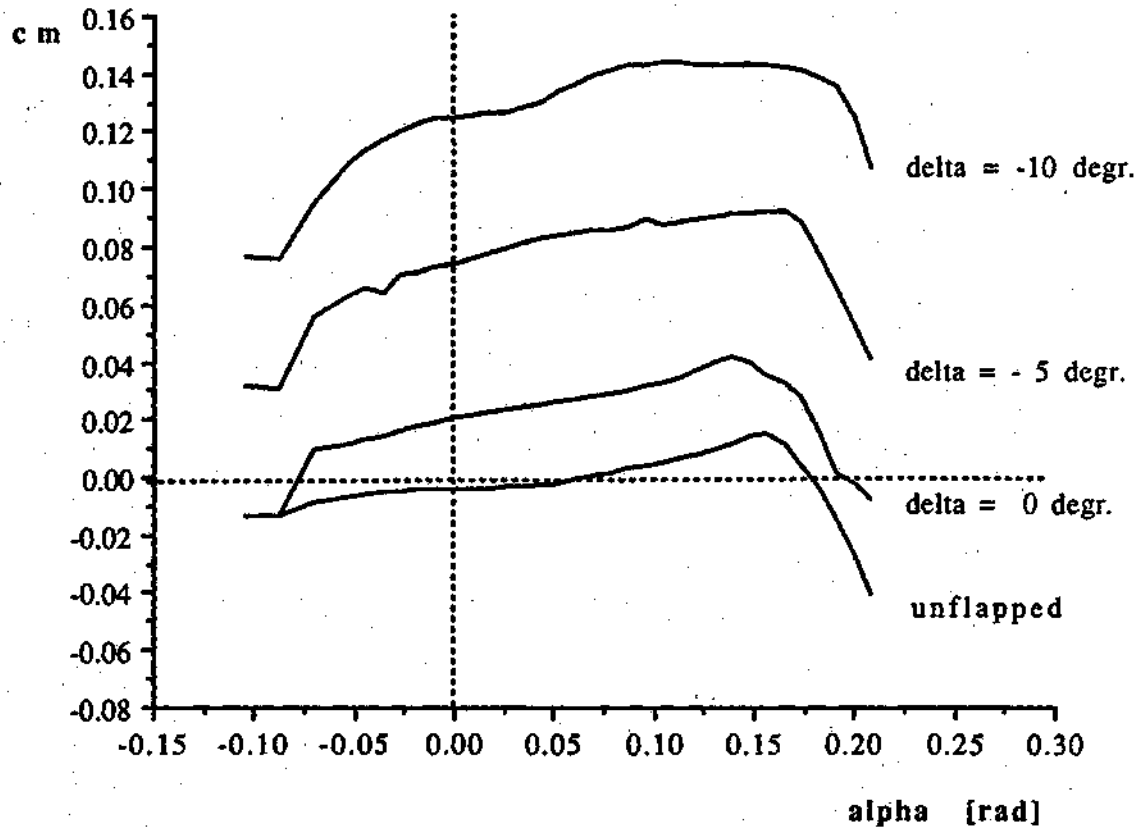


Fig. 9: Kaman 101 Rotor Blade and Blade/Flap Moment Coefficient, M = 0.5

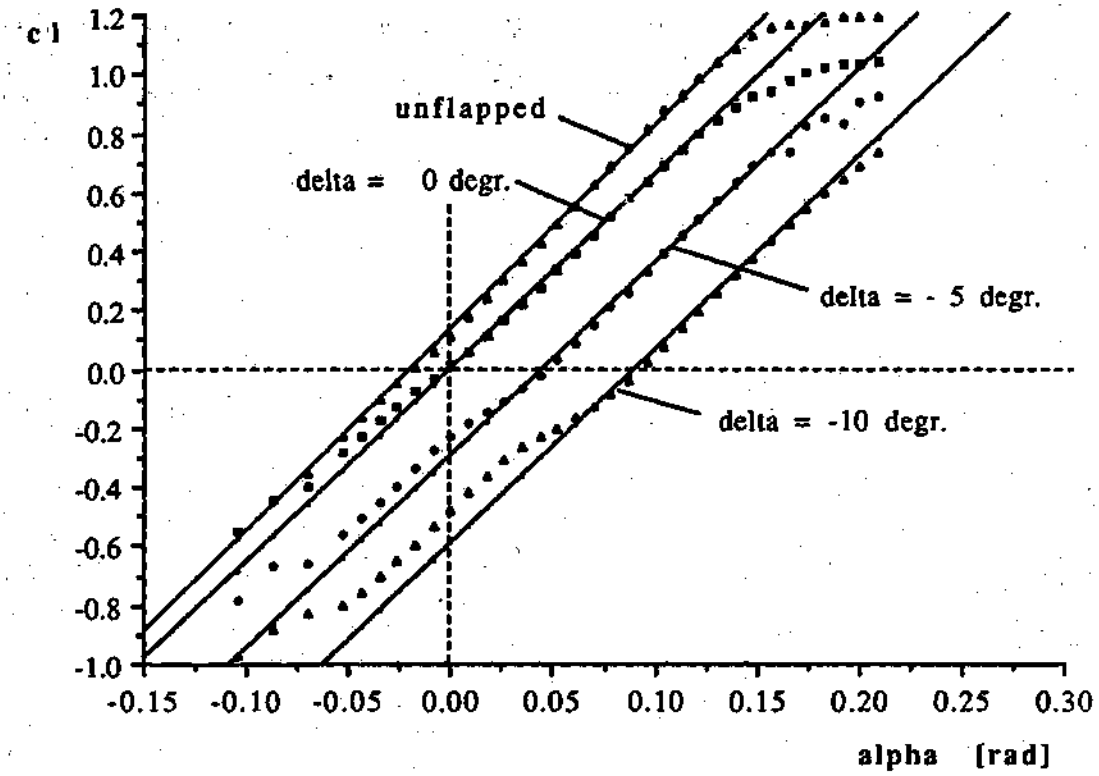


Fig. 10: Kaman Test Data and Linear Approximation (c_l):

$$c_l^b = 6.8390\,\alpha + 0.1367$$

$$c_l^{bf} = 6.5456\,\alpha + 3.372\,\delta$$

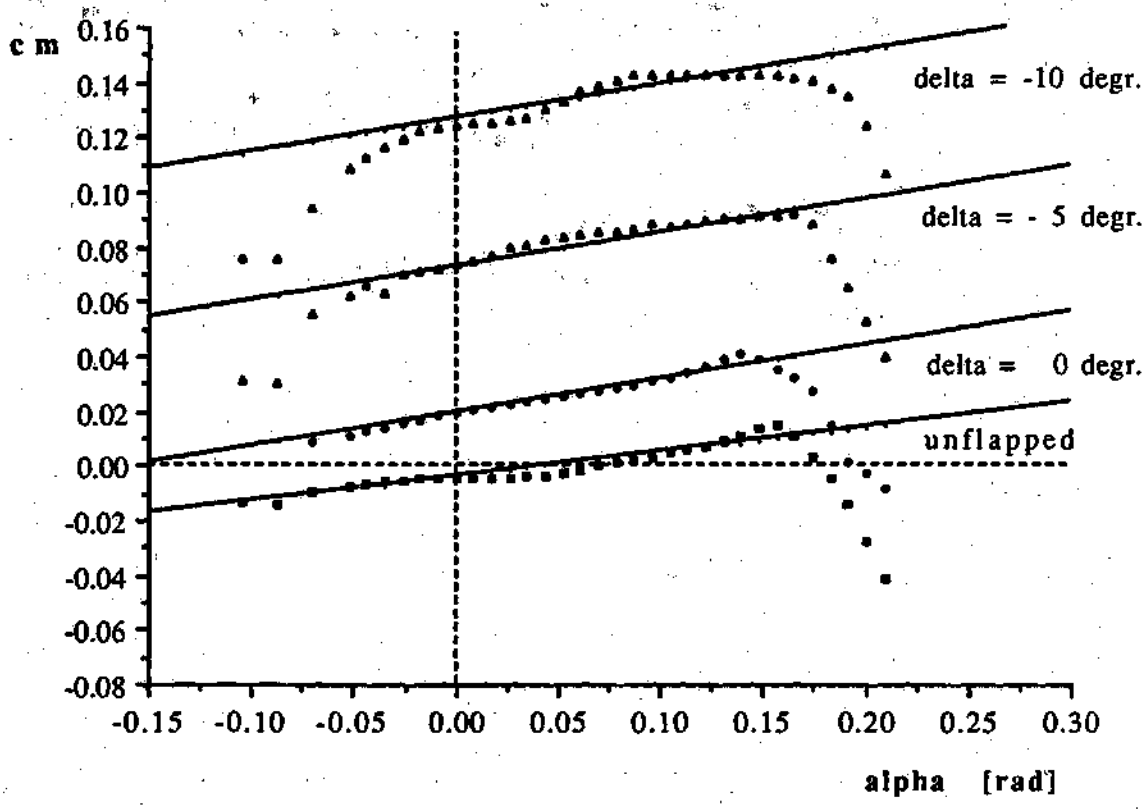
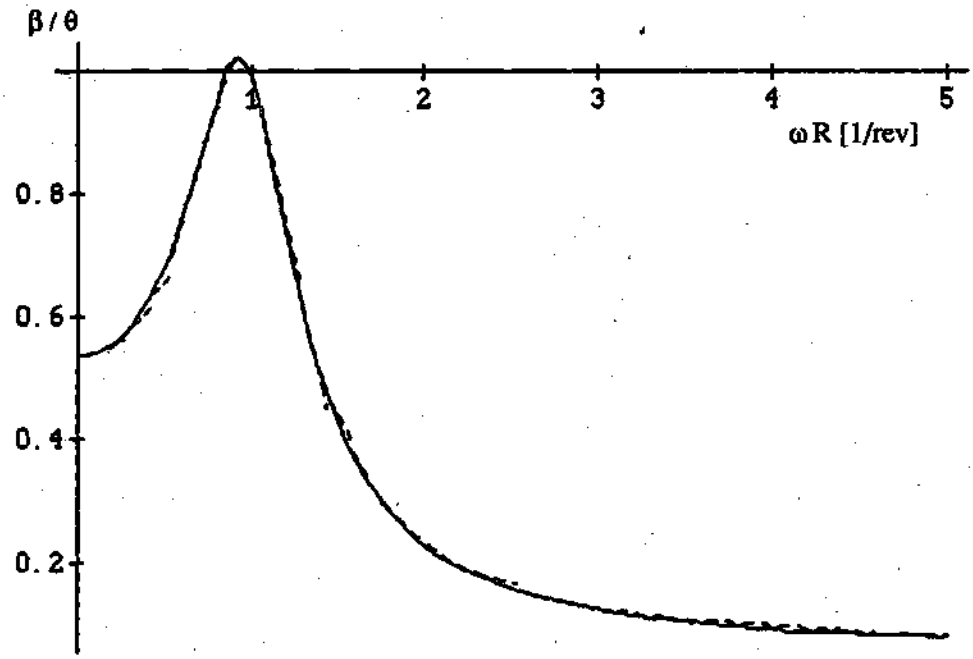


Fig. 11: **Kaman Test Data and Linear Approximation (c_m):**

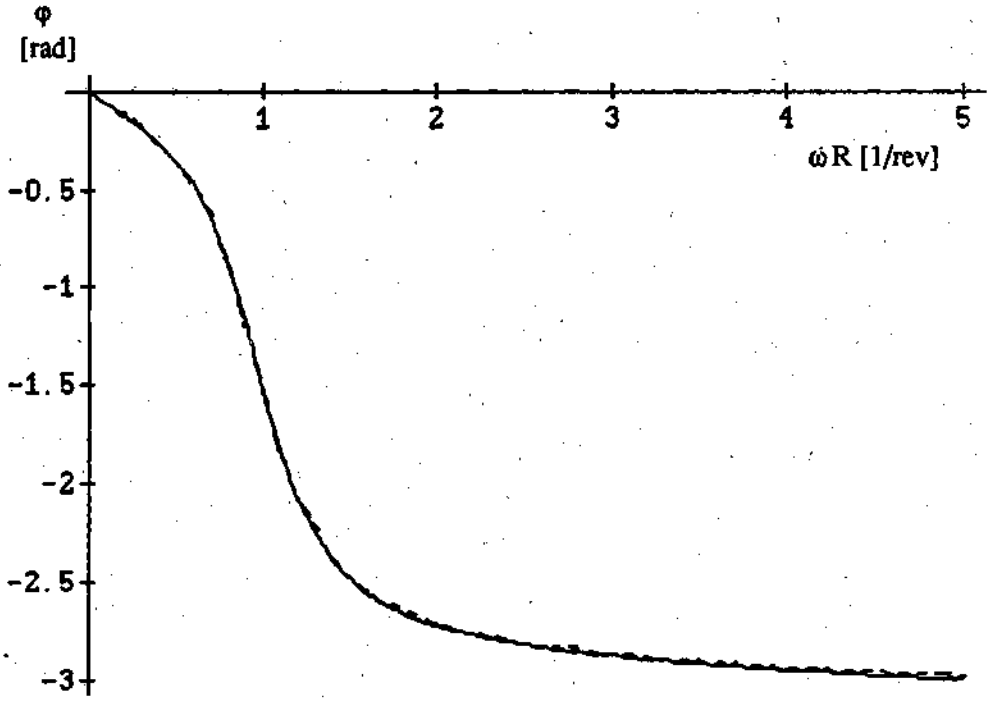
$$c_m{}^b = -0.004 + 0.0887\ \alpha$$

$$c_m{}^{bf} = 0.02 + 0.1234\ \alpha - 0.612\ \delta$$



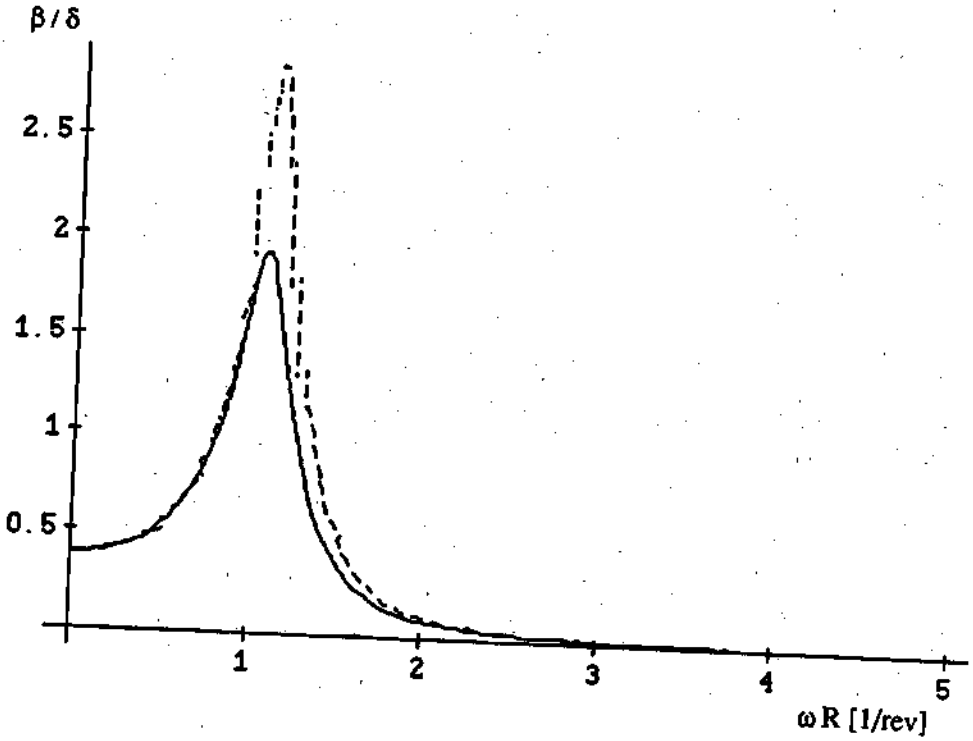
dashed lines: quasi-steady; solid lines: incl. apparent mass terms

Fig. 12a: Amplitude β(θ), System I, no Inflow Dynamics



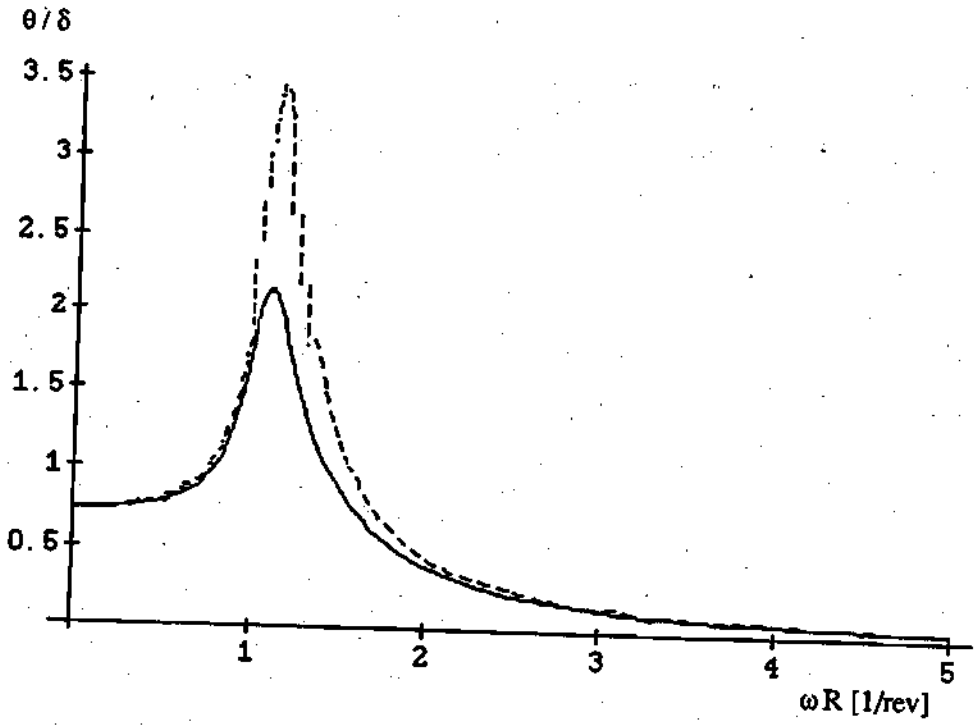
dashed lines: quasi-steady; solid lines: incl. apparent mass terms

Fig. 12b: Phase β(θ), System I, no Inflow Dynamics



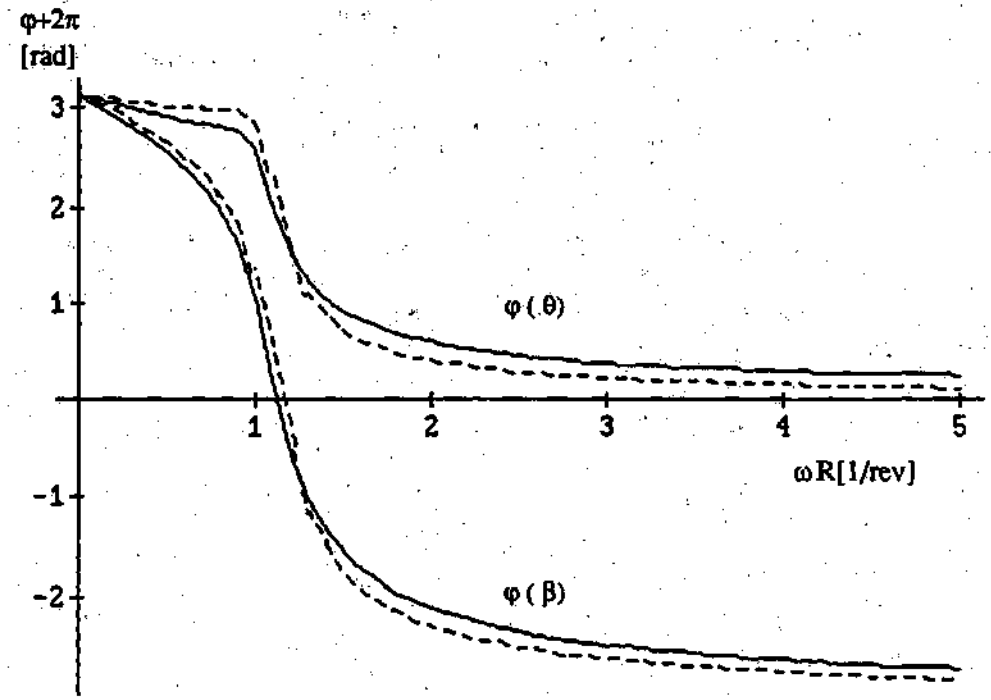
dashed lines: quasi-steady; solid lines: incl. apparent mass terms

Fig. 13a: Amplitude β(δ), System I, no Inflow Dynamics



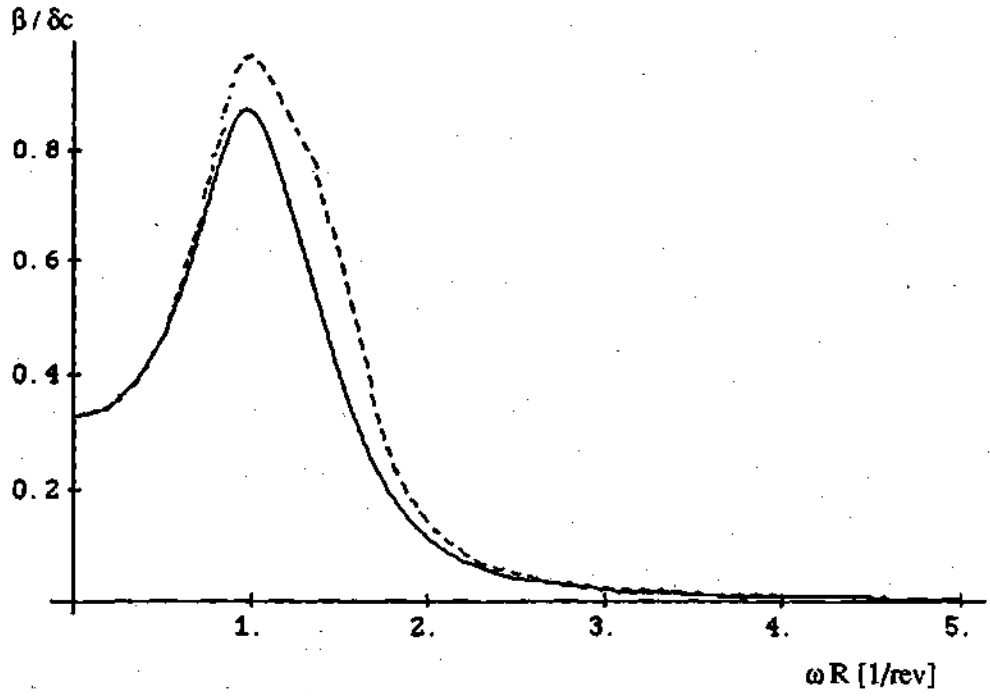
dashed lines: quasi-steady; solid lines: incl. apparent mass terms

Fig. 13b: Amplitude θ(δ), System I, no Inflow Dynamics



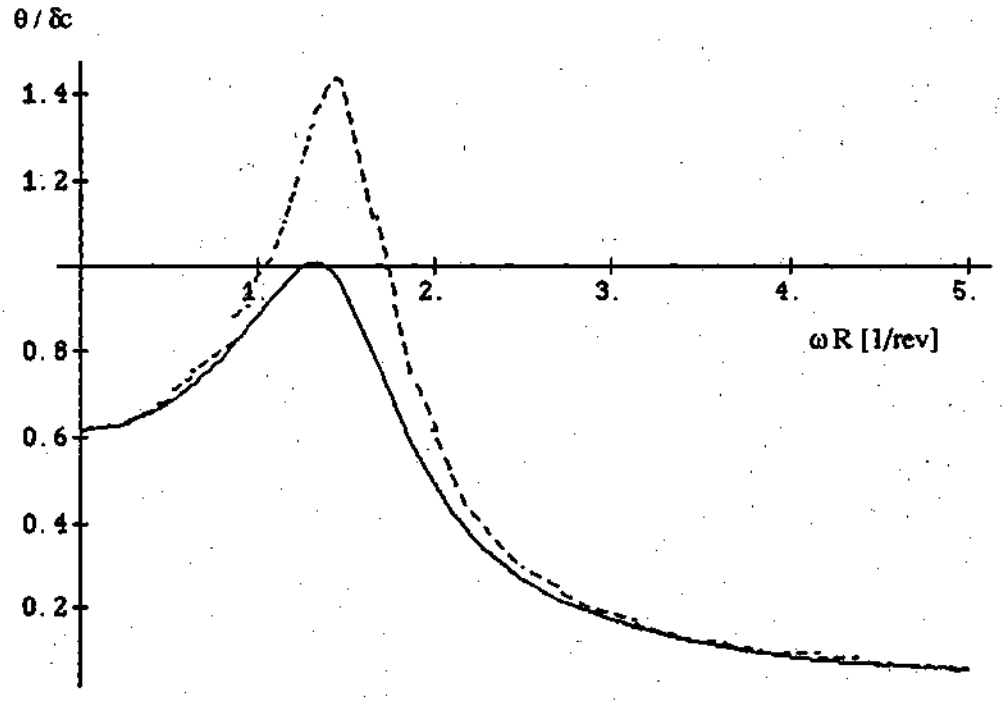
dashed lines: quasi-steady; solid lines: incl. apparent mass terms

Fig. 13c: Phase β(δ) and θ(δ), System I, no Inflow Dynamics



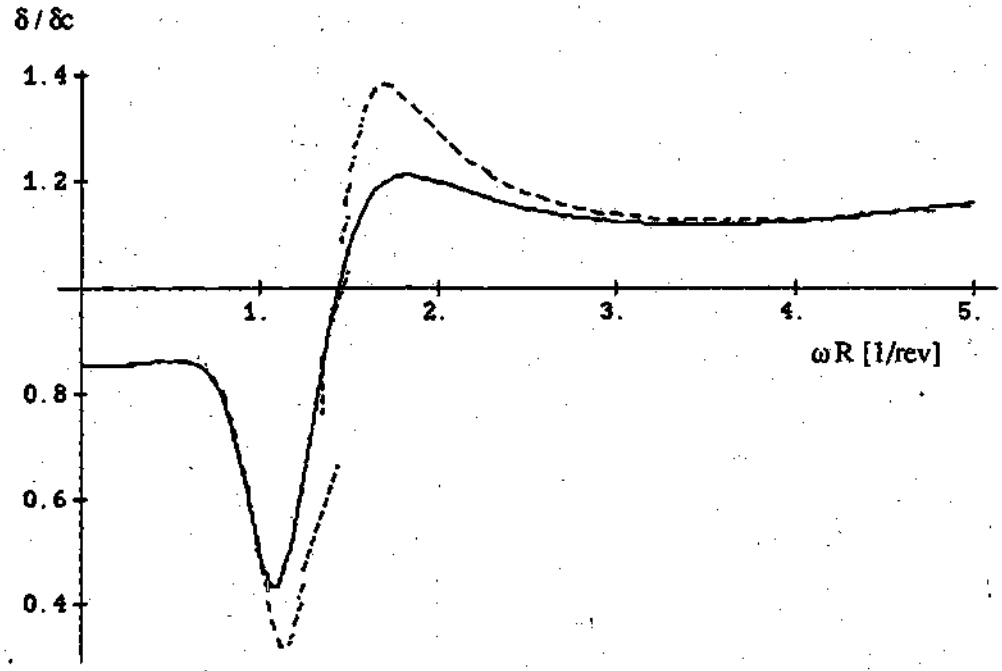
dashed lines: quasi-steady; solid lines: incl. apparent mass terms

Fig. 14a: Amplitude $\beta(\delta_c)$, System II, no Inflow Dynamics



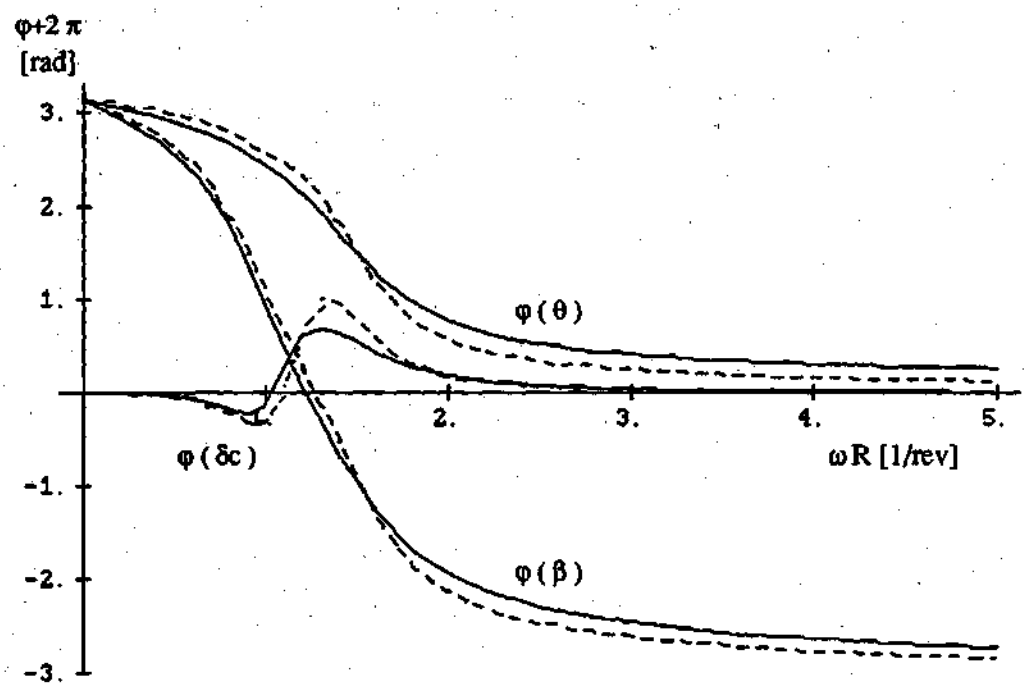
dashed lines: quasi-steady; solid lines: incl. apparent mass terms

Fig. 14b: Amplitude $\theta(\delta_c)$, System II, no Inflow Dynamics



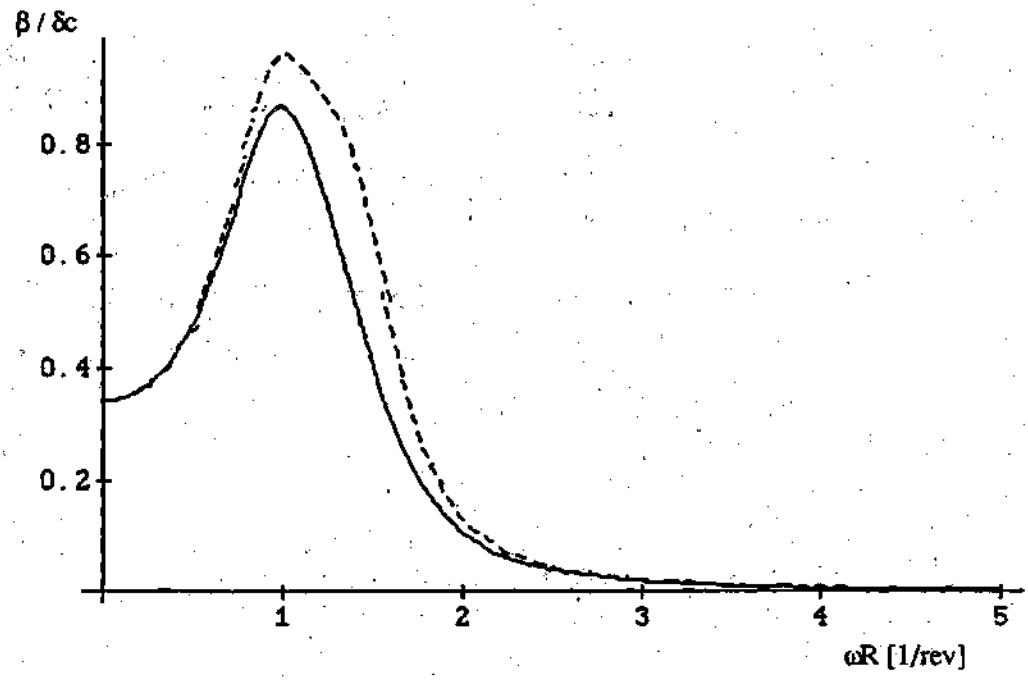
dashed lines: quasi-steady; solid lines: incl. apparent mass terms

Fig. 14c: Amplitude $\delta(\delta_c)$, System II, no Inflow Dynamics



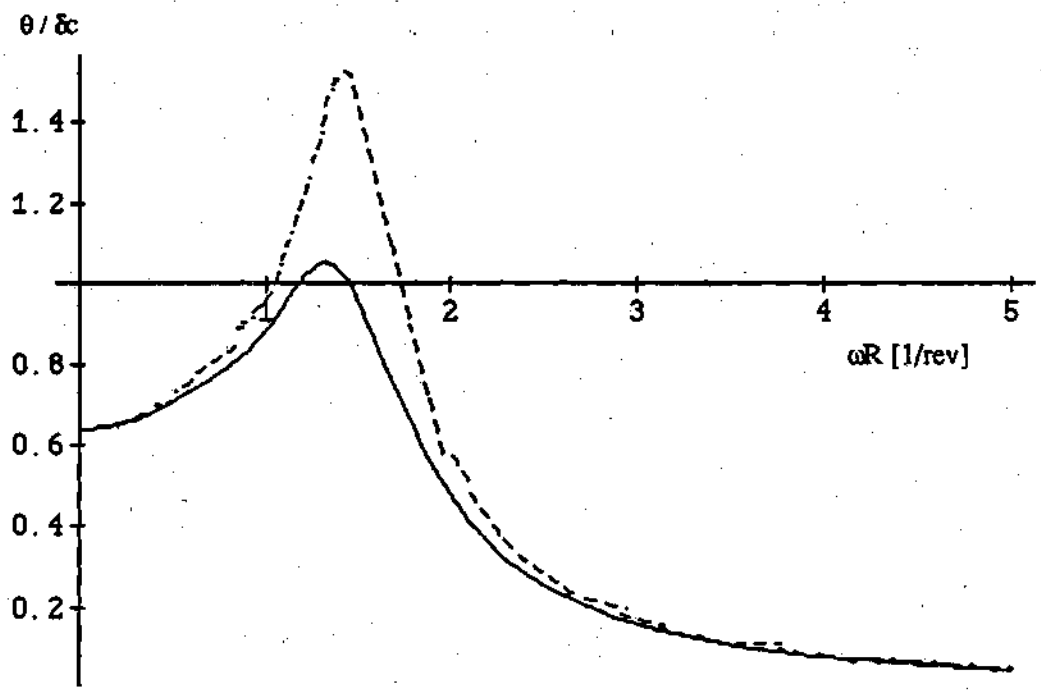
dashed lines: quasi-steady; solid lines: incl. apparent mass terms

Fig. 14d: Phase $\beta(\delta_c)$, $\theta(\delta_c)$, and $\delta(\delta_c)$, System II, no Inflow Dynamics



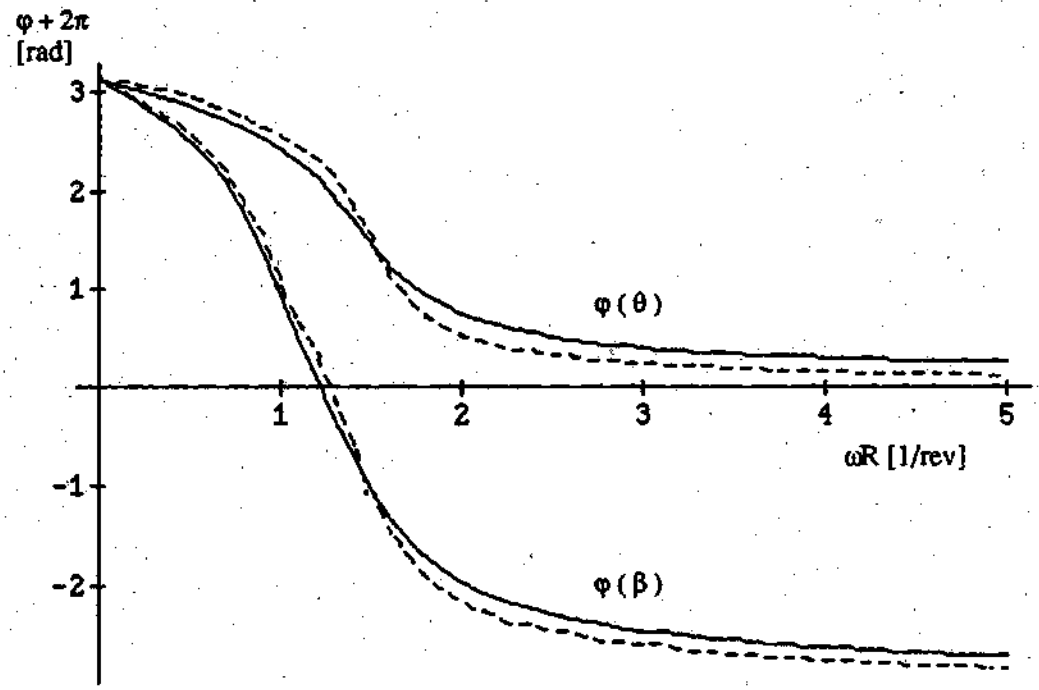
dashed lines: quasi-steady; solid lines: incl. apparent mass terms

Fig. 15a: Amplitude $\beta(\delta_c)$, System III, no Inflow Dynamics



dashed lines: quasi-steady; solid lines: incl. apparent mass terms

Fig. 15b: Amplitude $\theta(\delta_c)$, System III, no Inflow Dynamics



dashed lines: quasi-steady; solid lines: incl. apparent mass terms

Fig. 15c: Phase $\beta(\delta_c)$ and $\theta(\delta_c)$, System III, no Inflow Dynamics

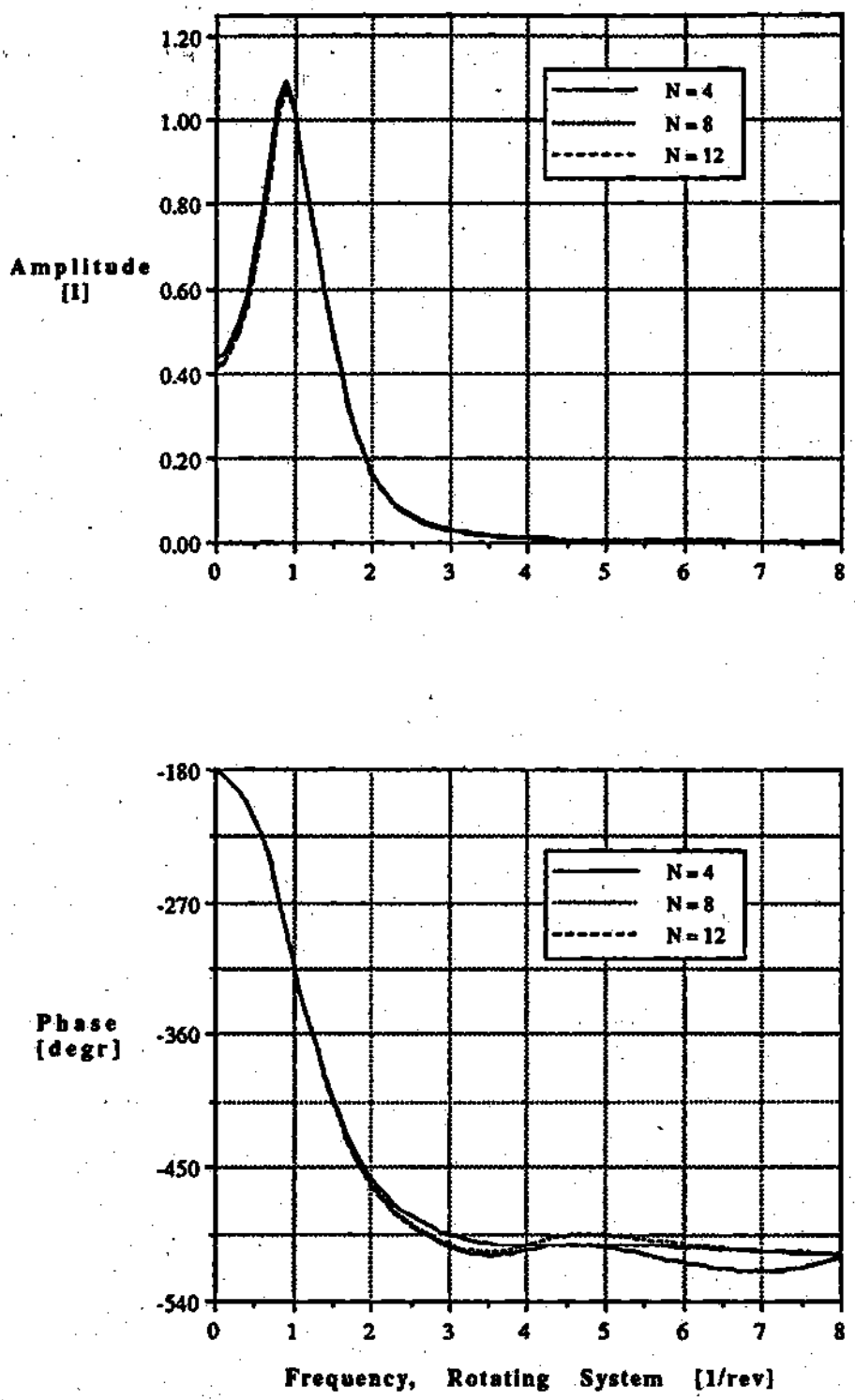


Fig. 16a: Flapping Response, Collective HHC Mode

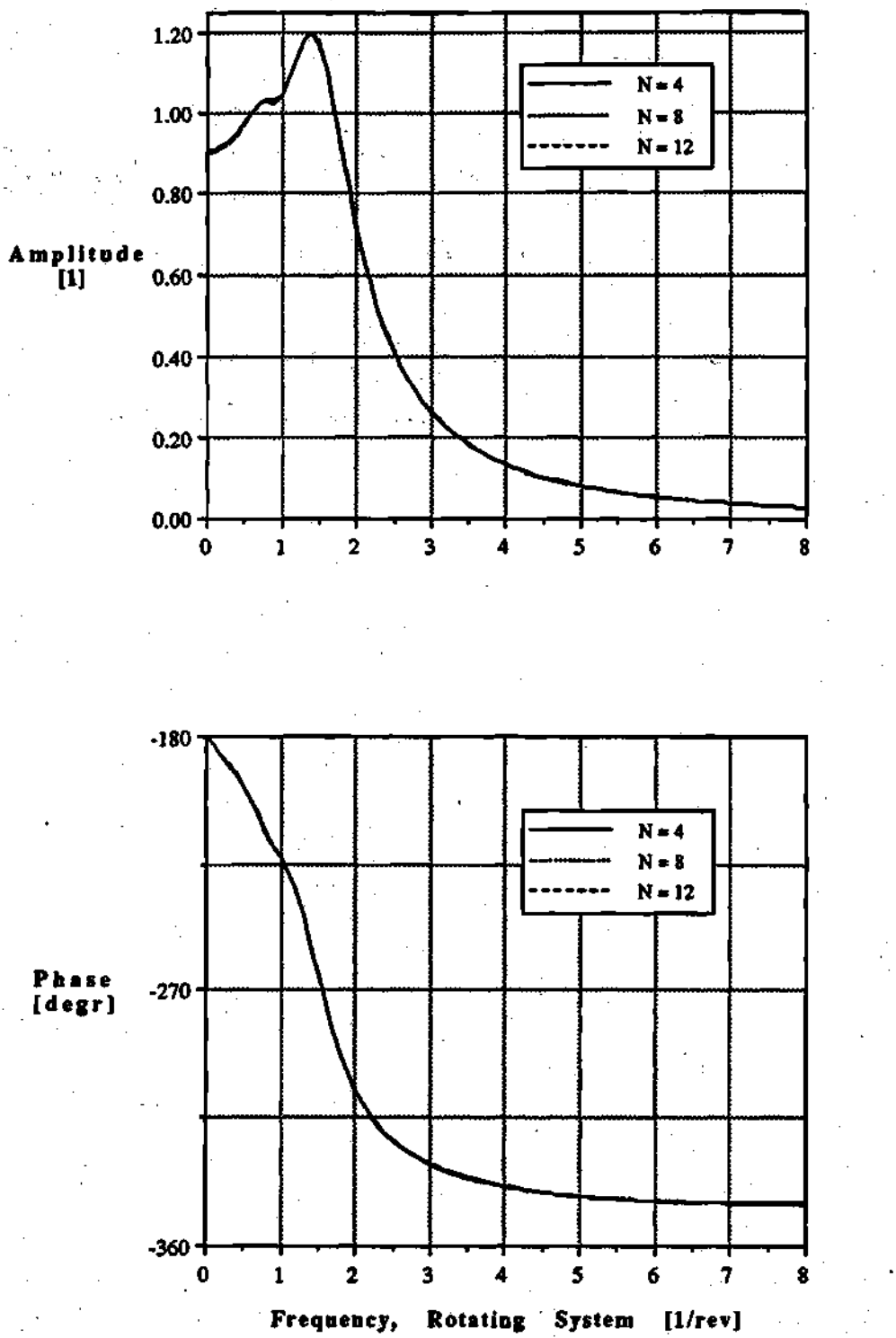


Fig. 16b: Feathering Response, Collective HHC Mode

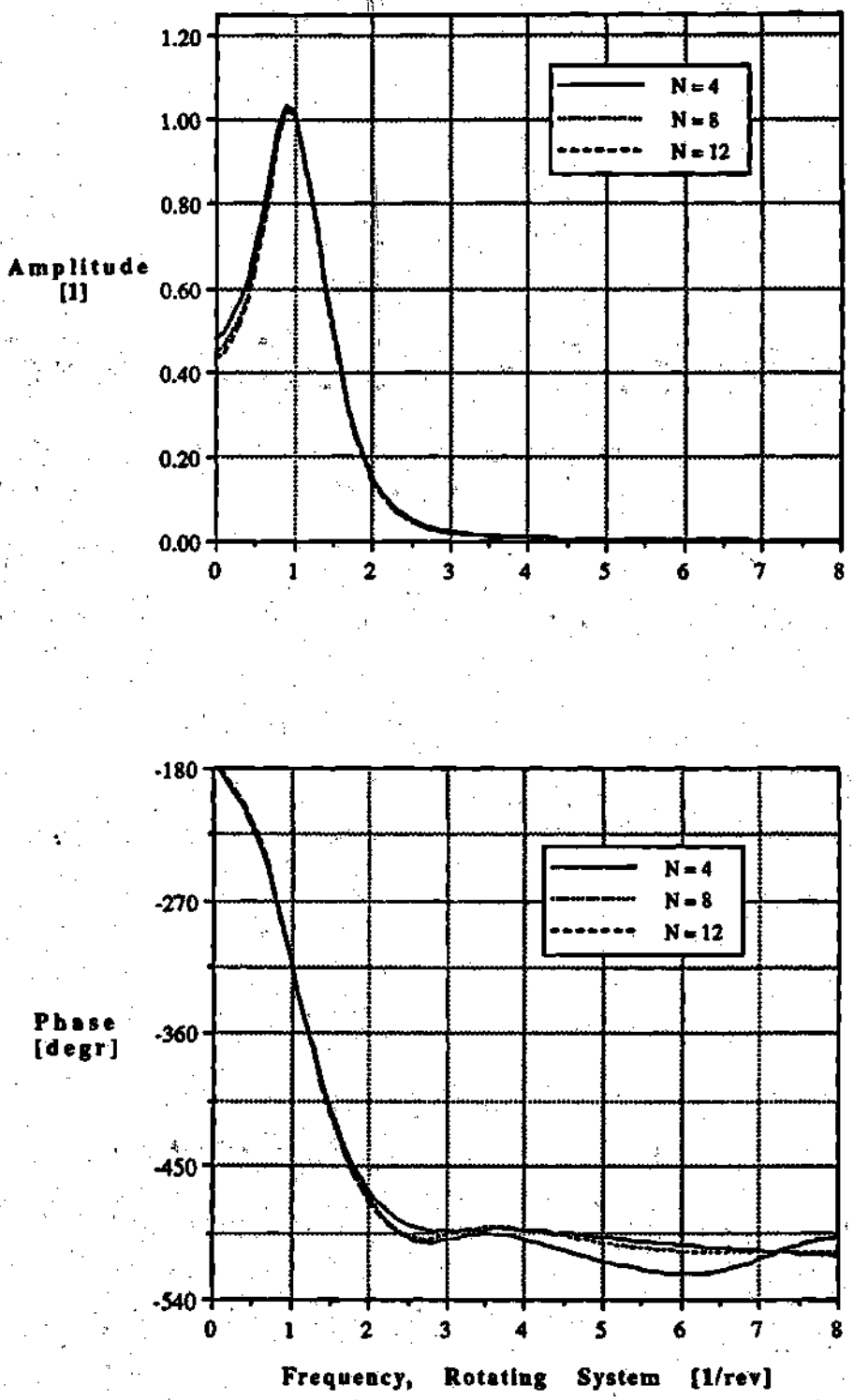


Fig. 17a: Flapping Response, Progressing HHC Mode

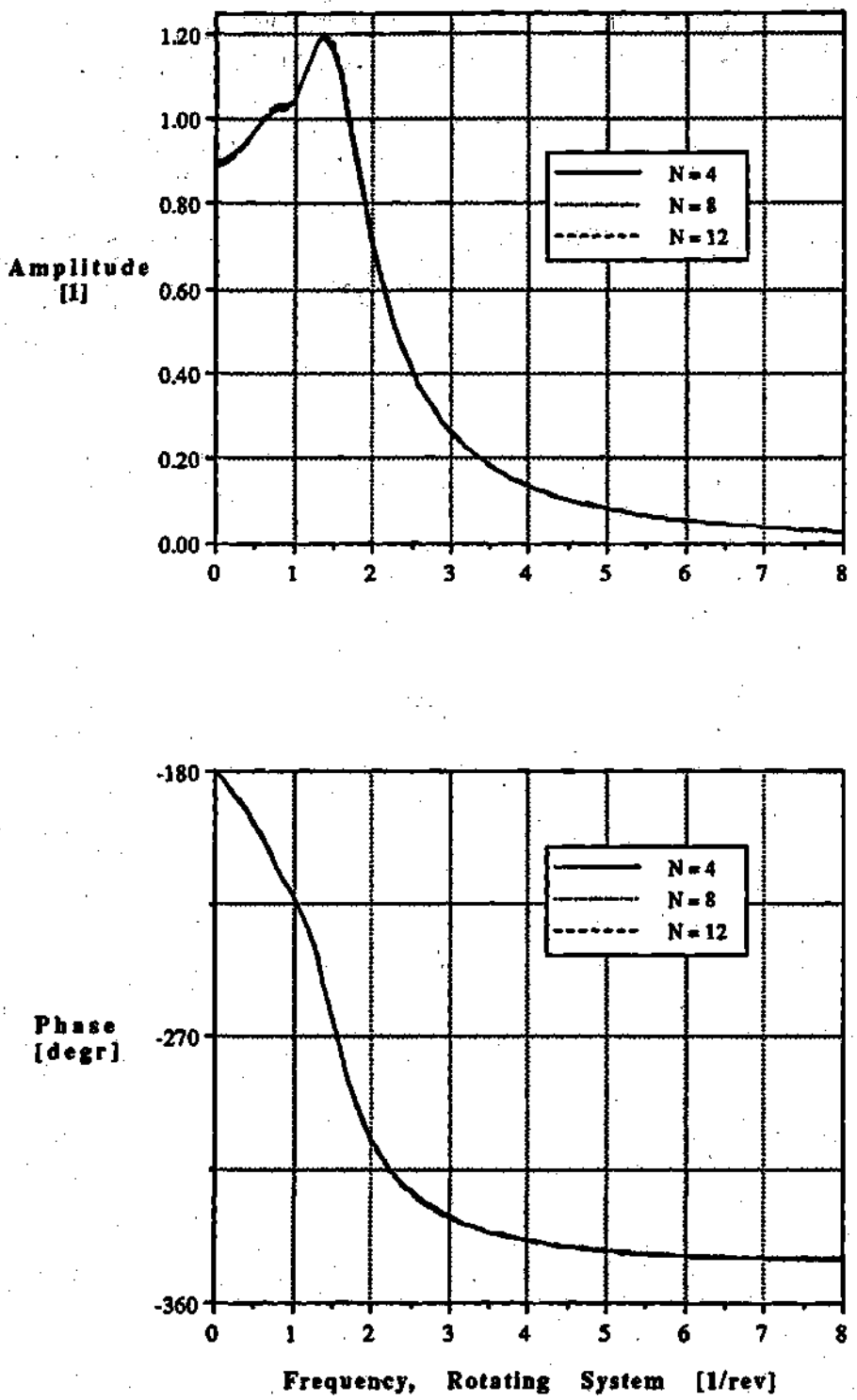


Fig. 17b: Feathering Response, Progressing HHC Mode

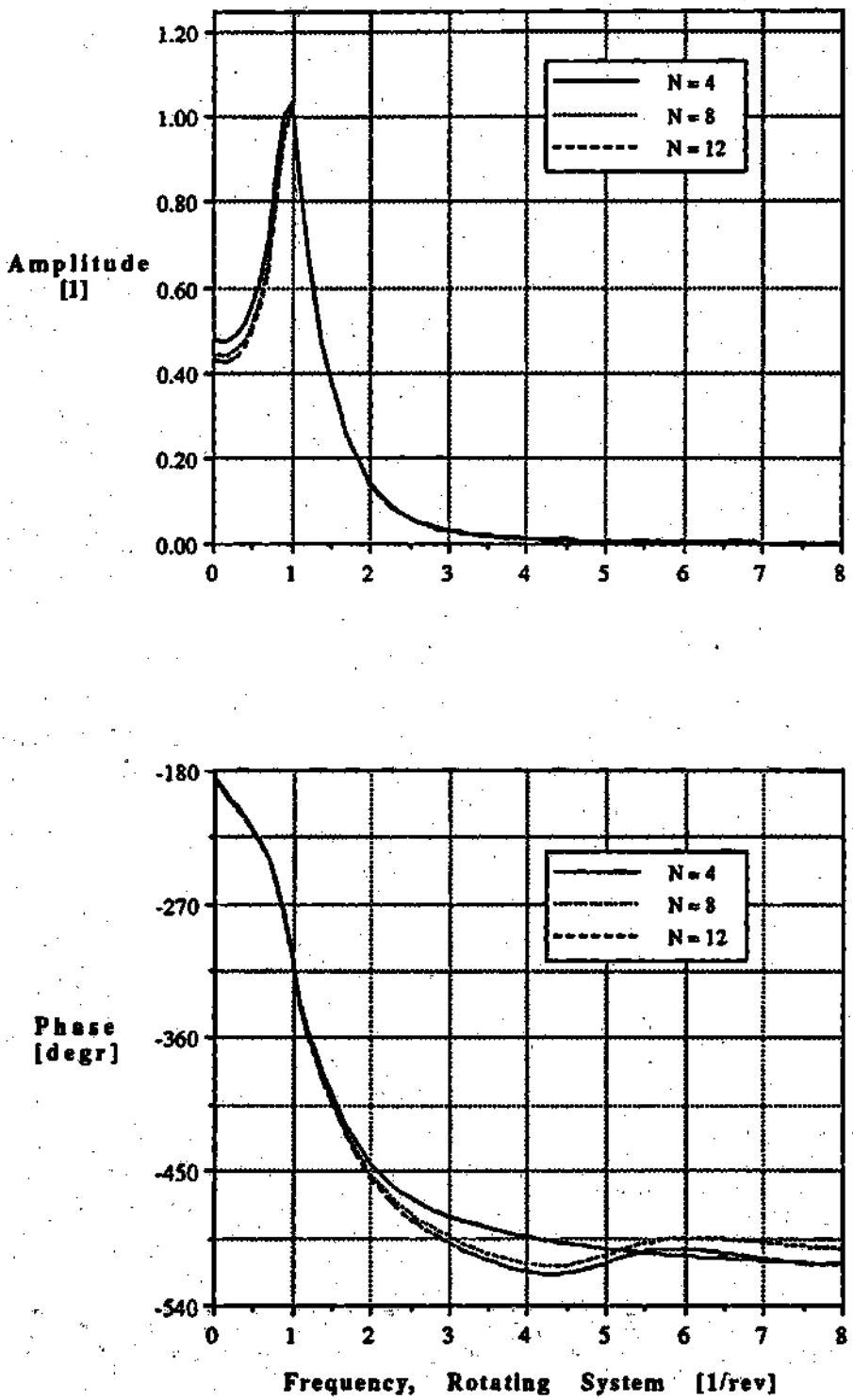


Fig. 18a: Flapping Response, Regressing HHC Mode

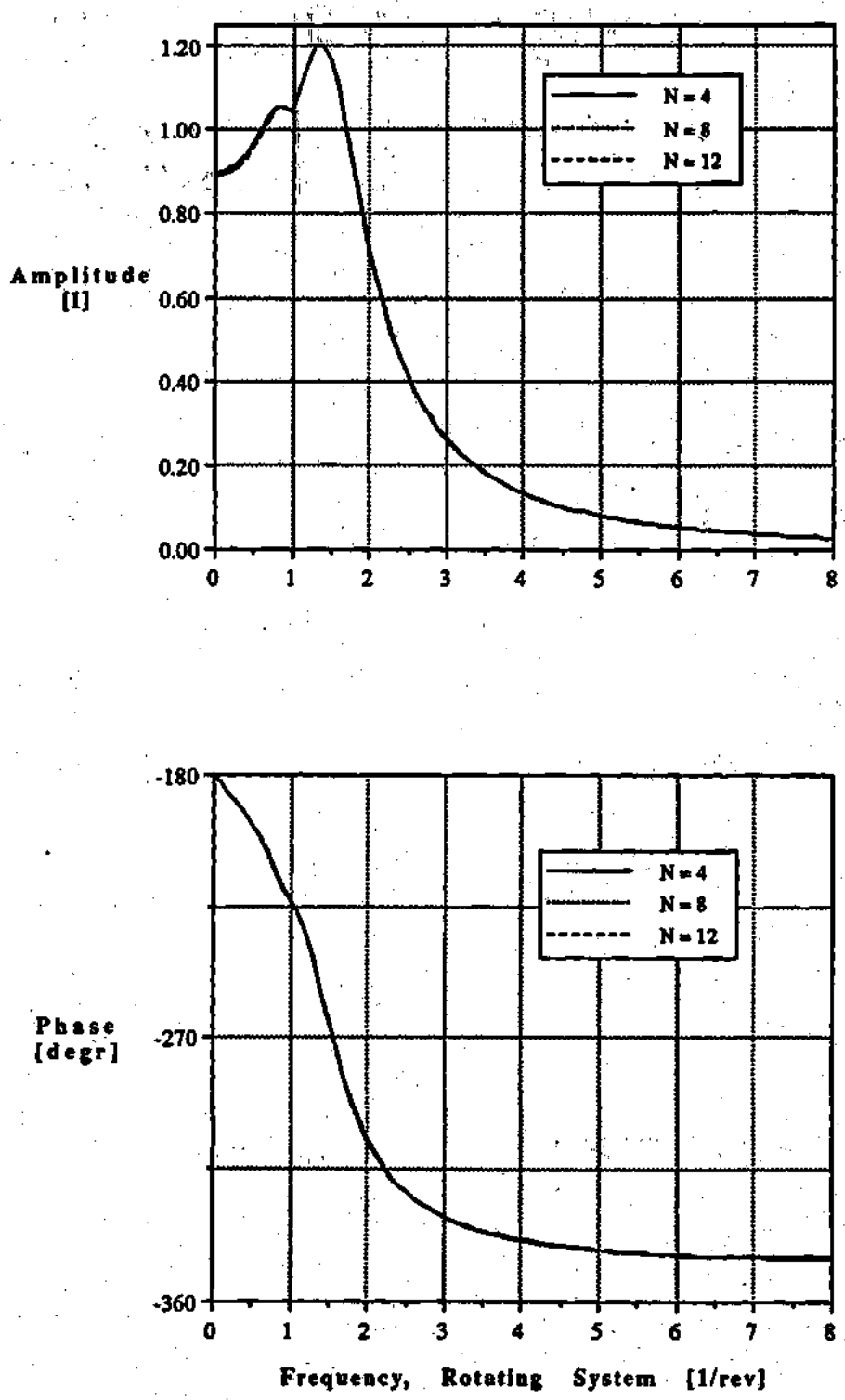


Fig. 18b: Feathering Response, Regressing HHC Mode

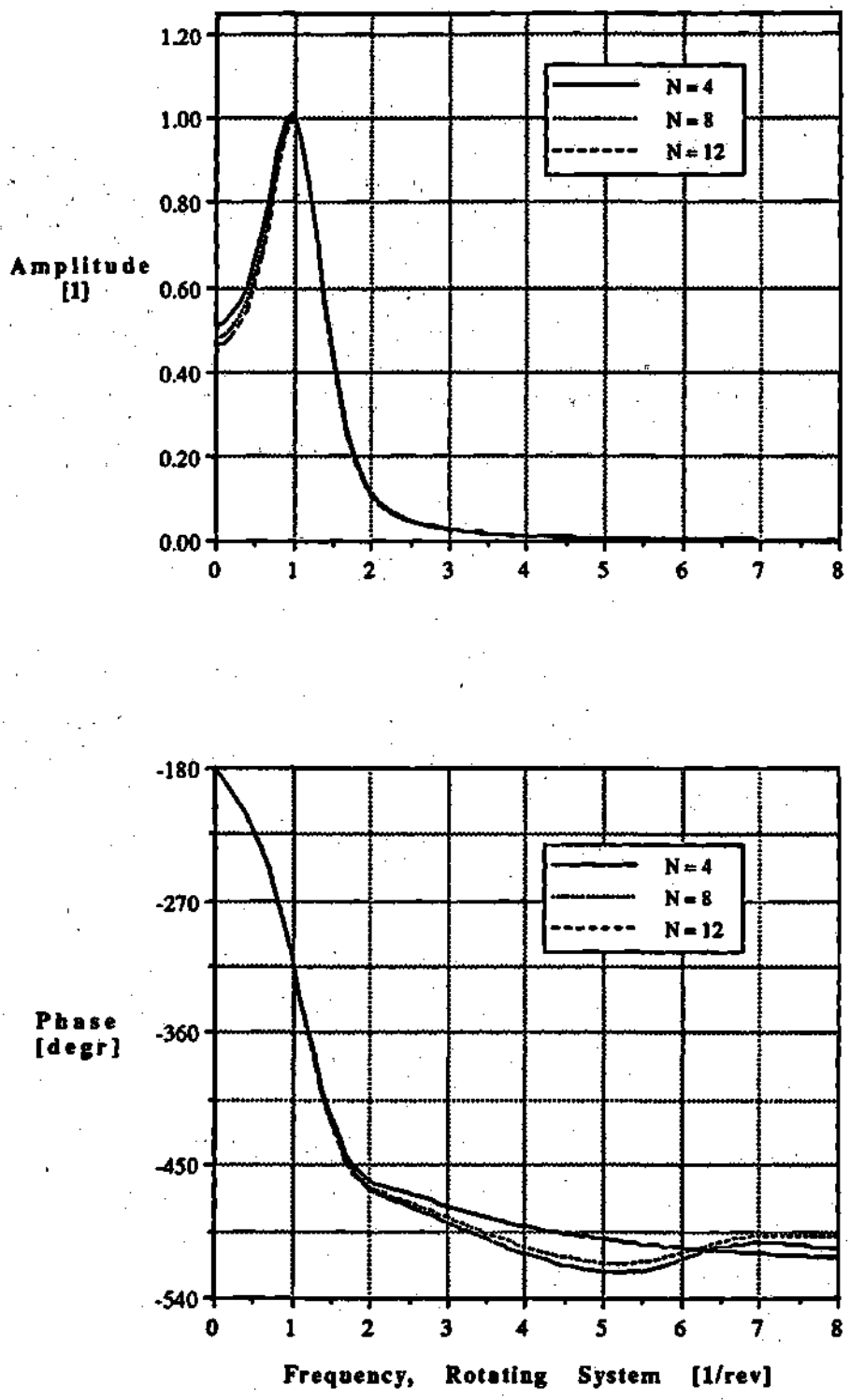


Fig. 19a: Flapping Response, Differential HHC Mode

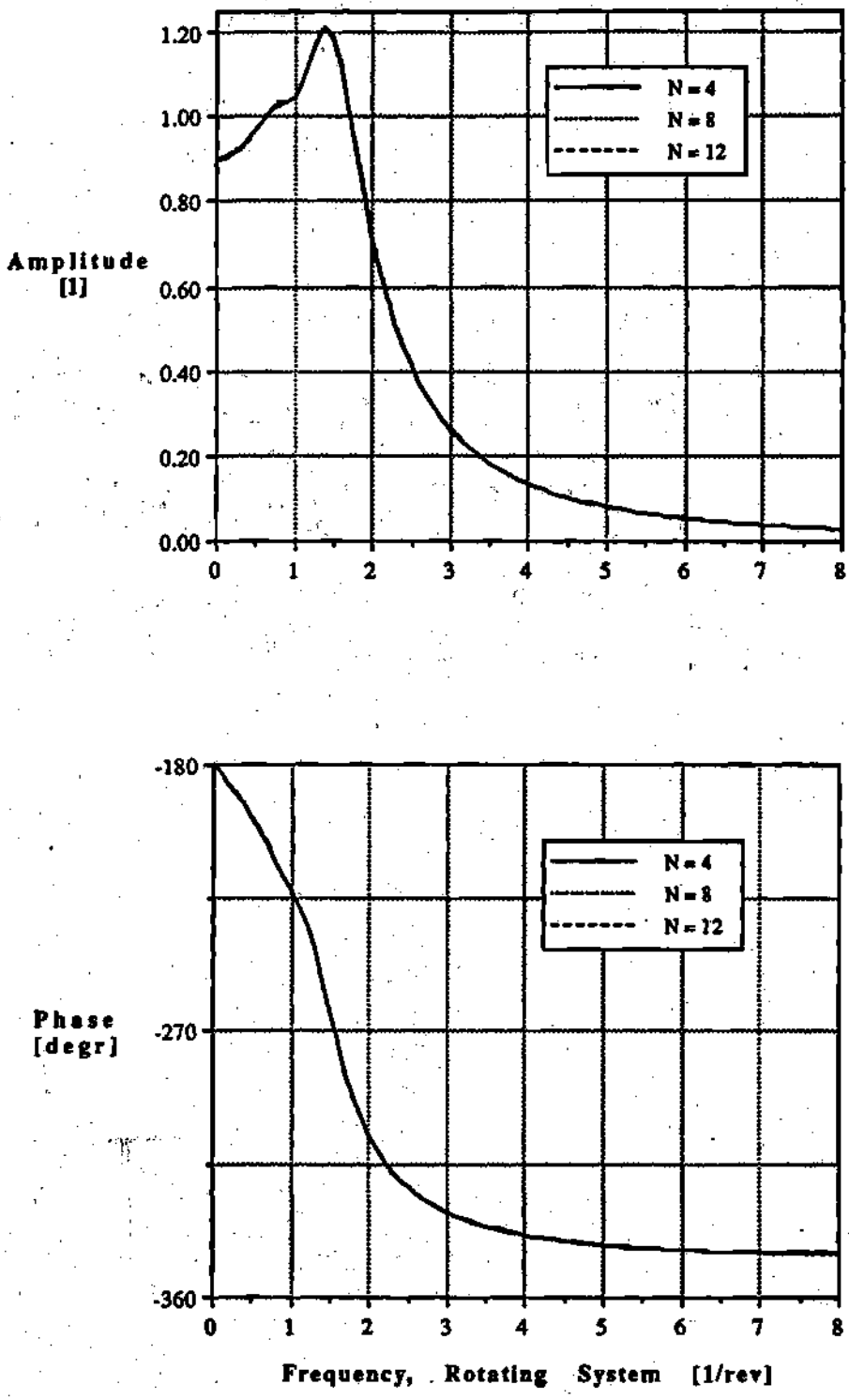
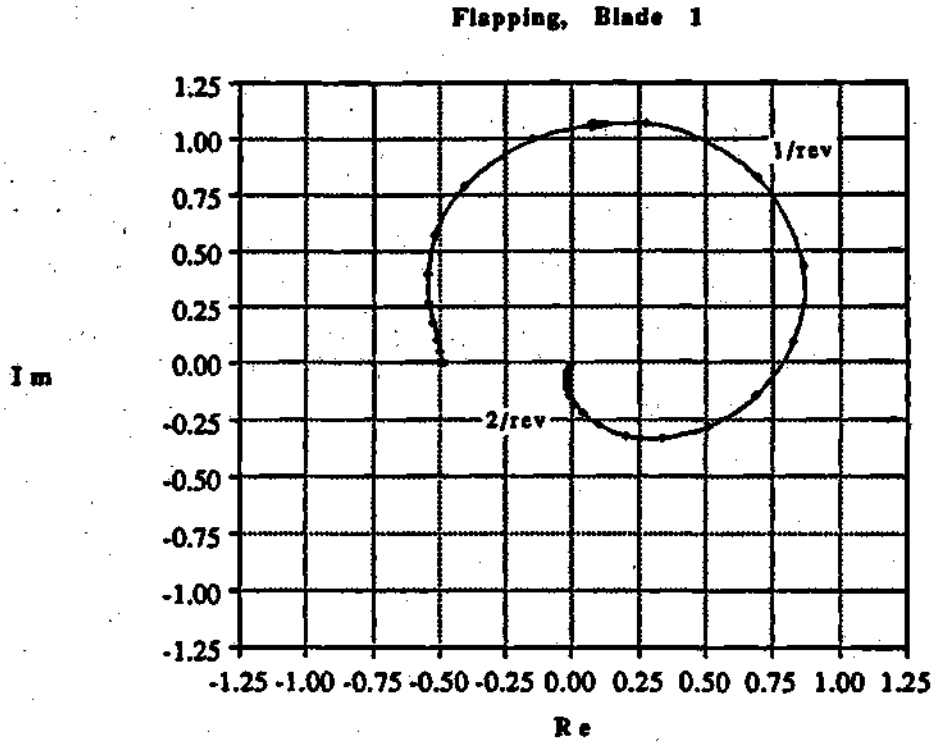


Fig. 19b: Feathering Response, Differential HHC Mode



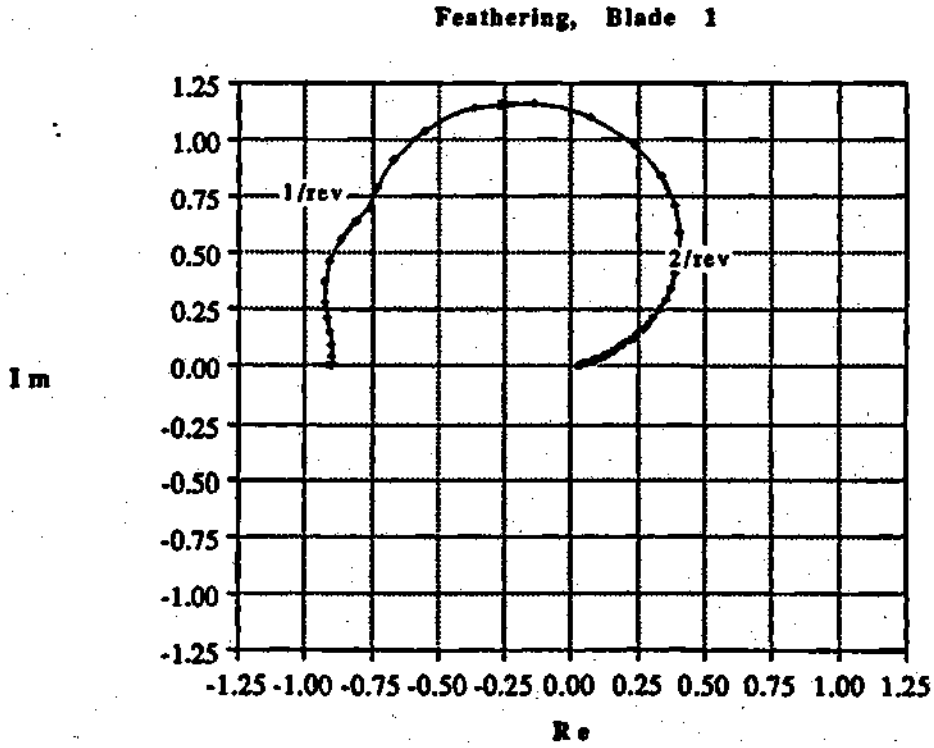


Fig. 20a: Nyquist Plots, Blade 1

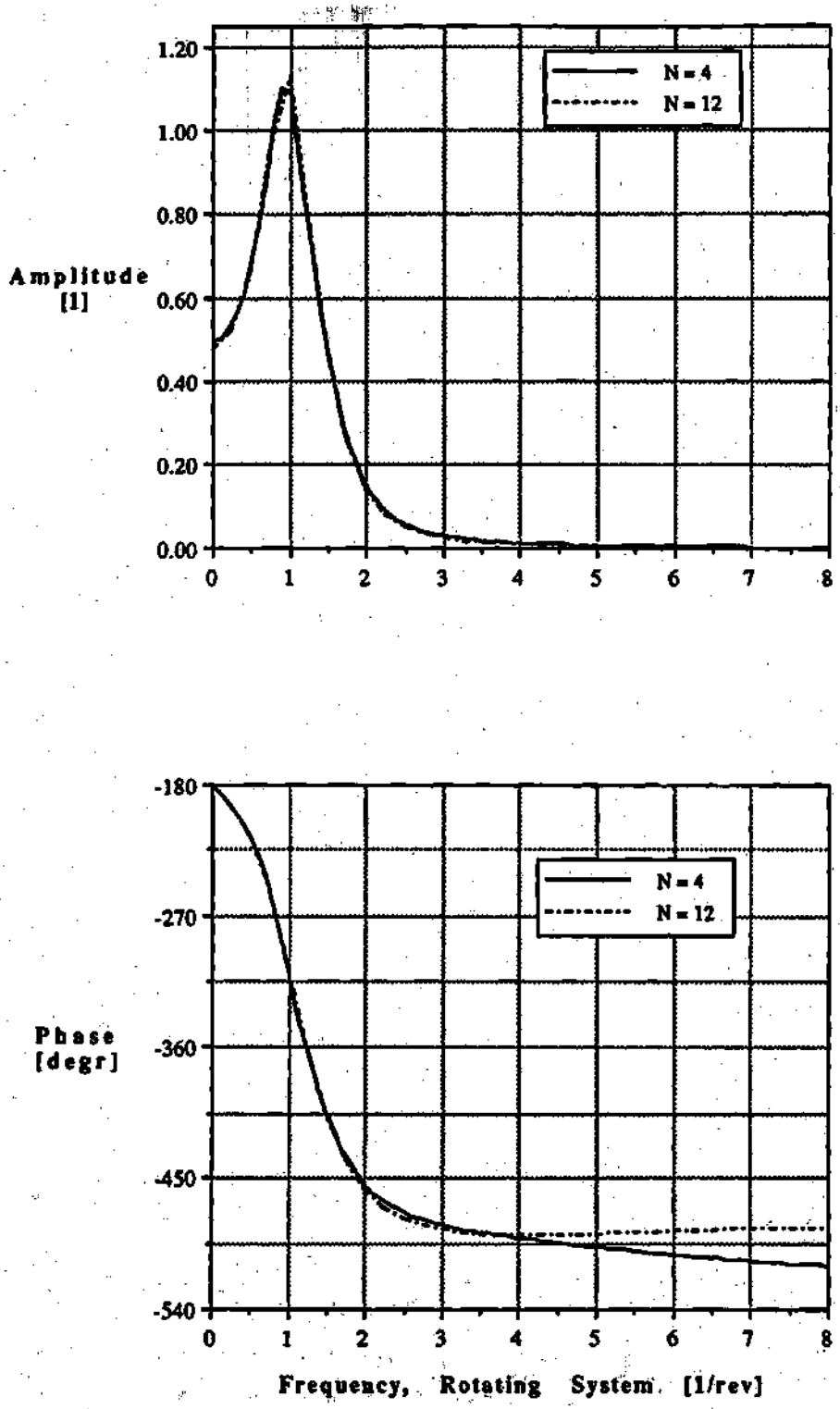


Fig. 20b: Flapping Response, Blade 1

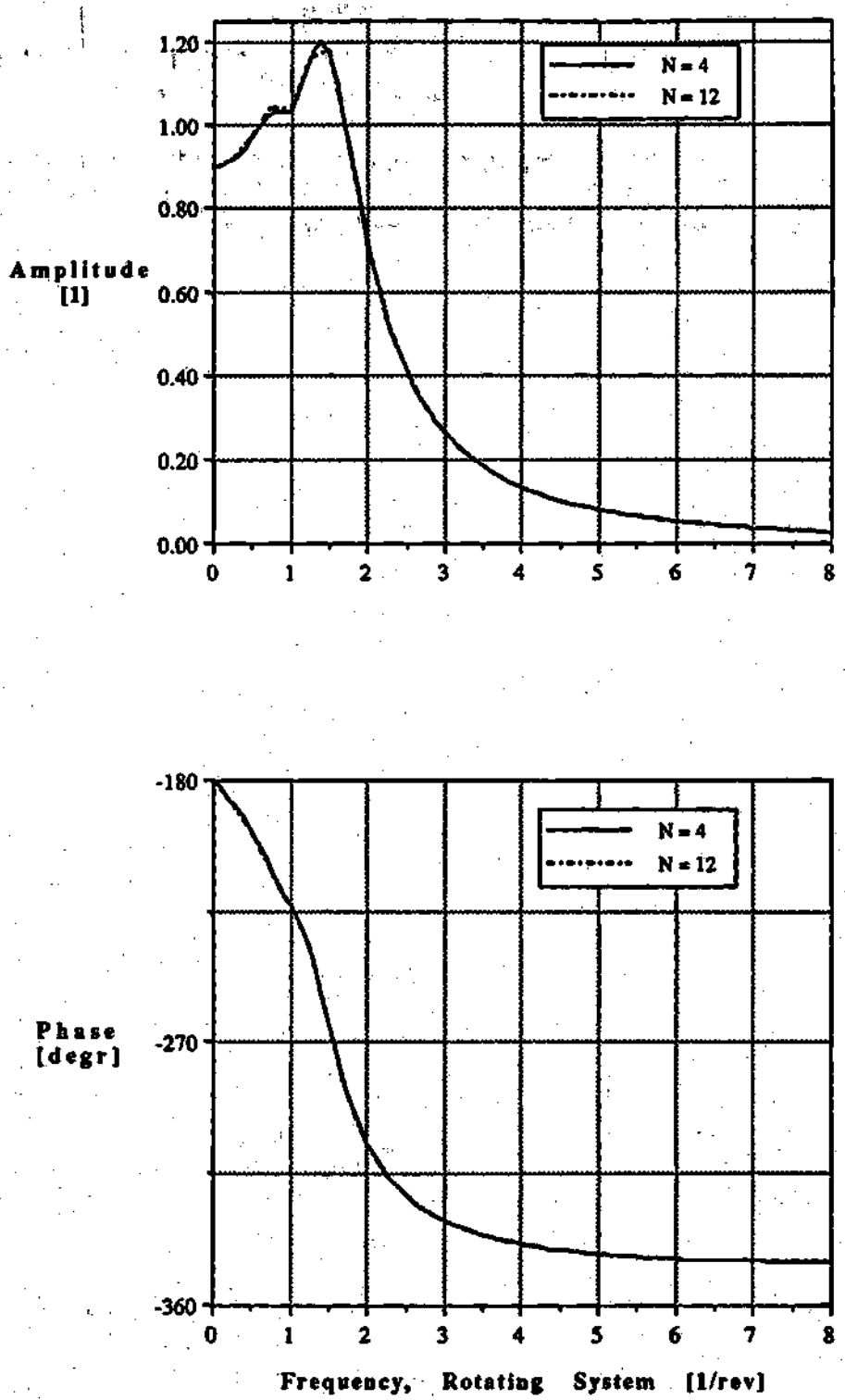
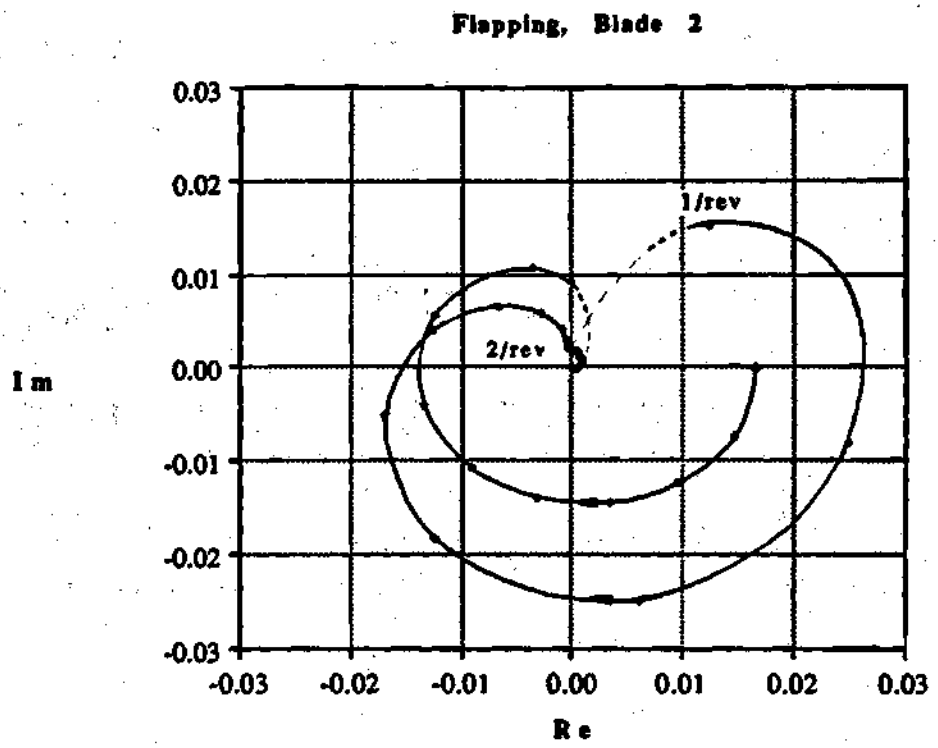


Fig. 20c: Feathering Response, Blade 1



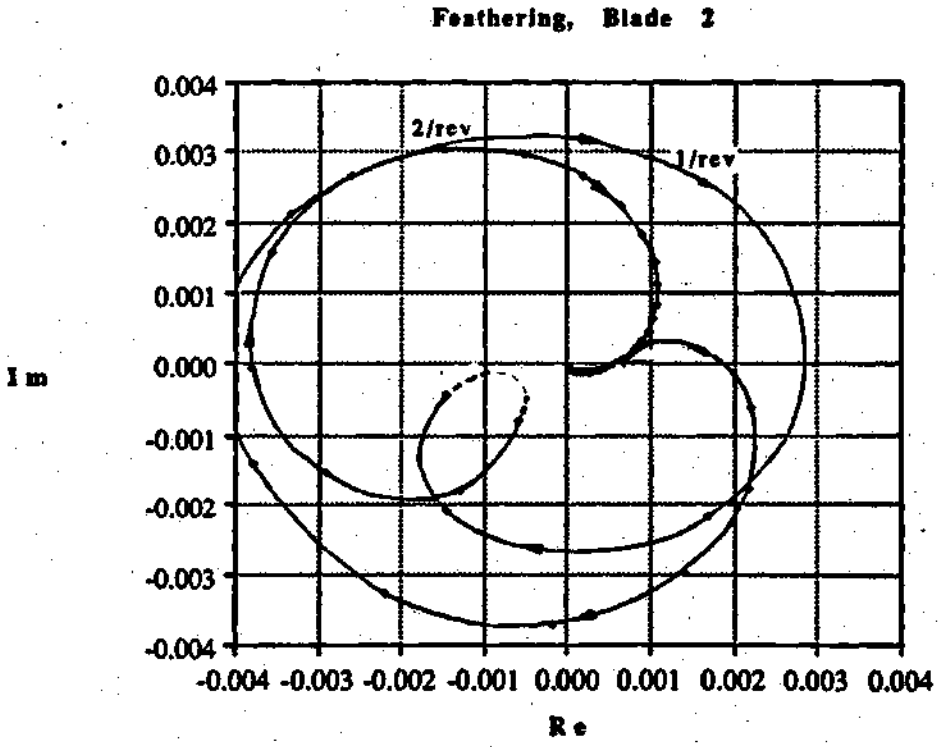


Fig. 21a: Nyquist Plots, Blade 2

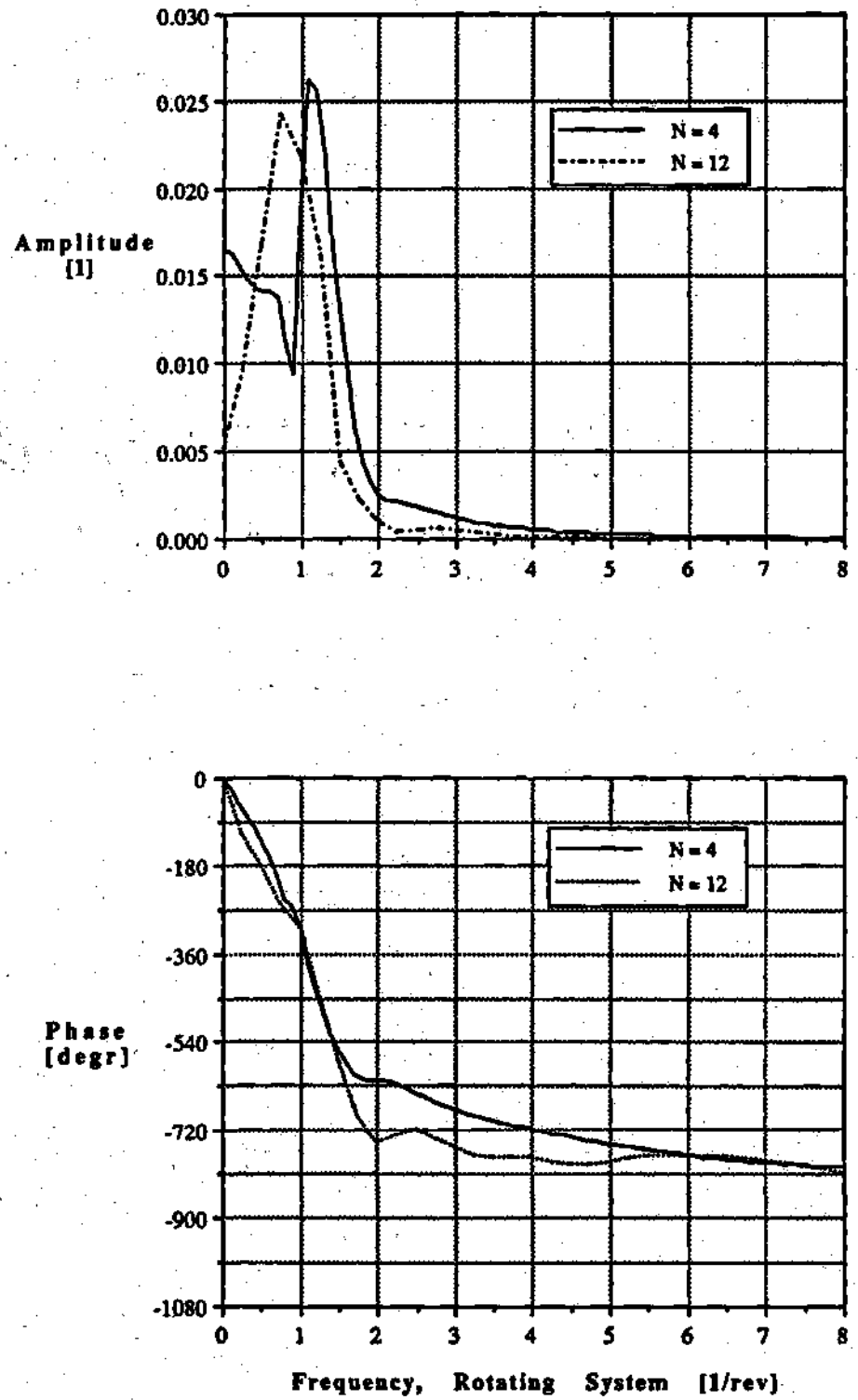


Fig. 21b: Flapping Response, Blade 2

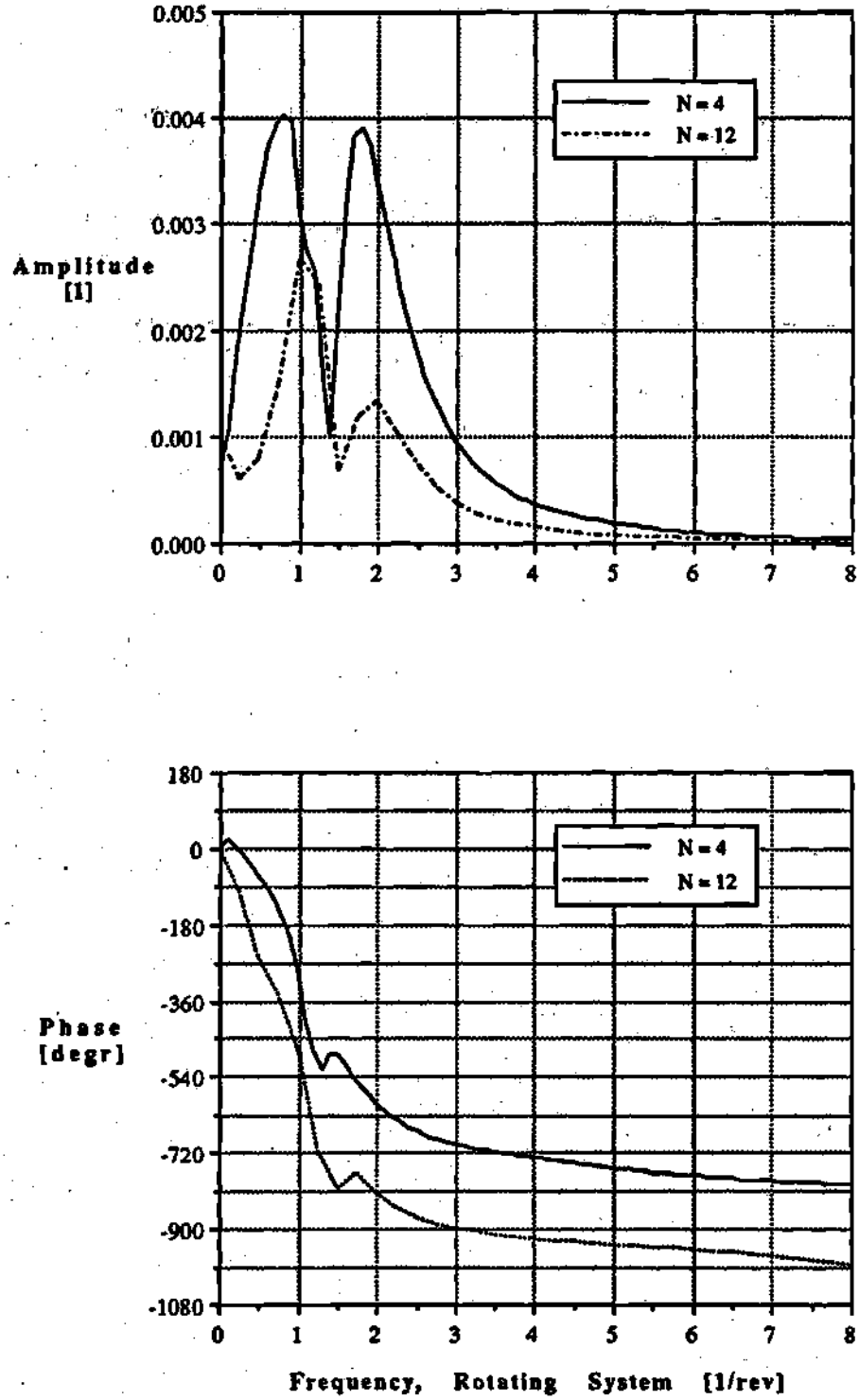
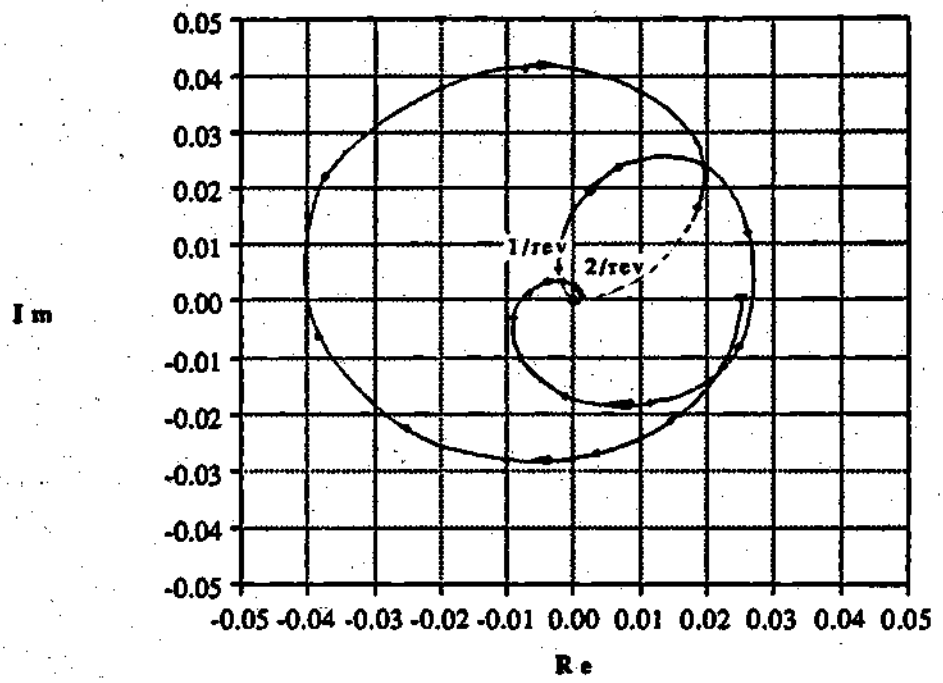


Fig. 21c: Feathering Response, Blade 2



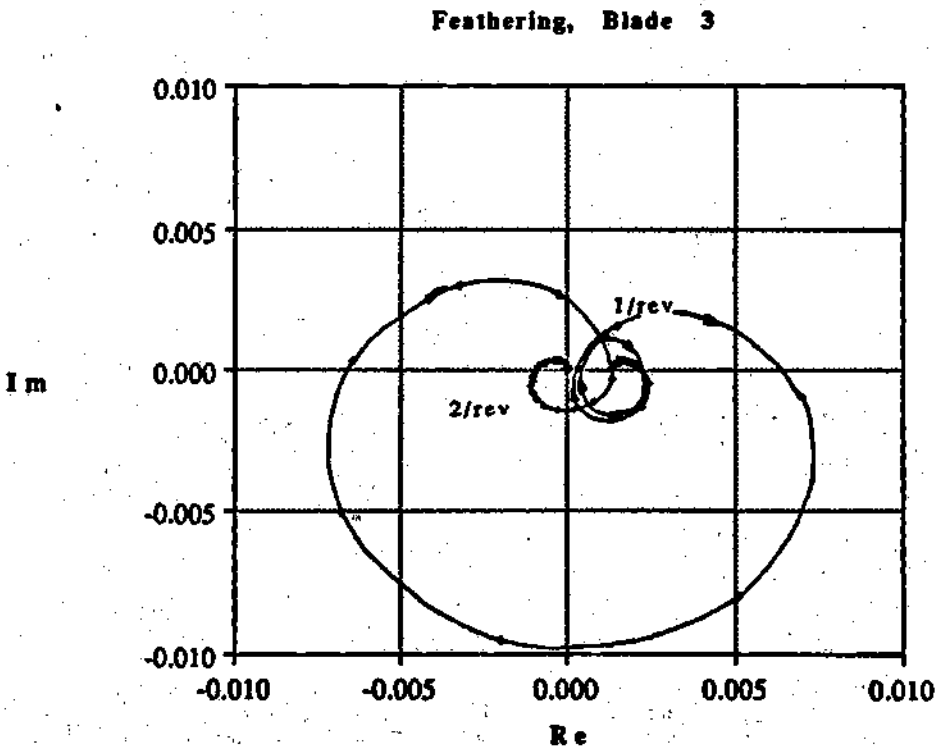


Fig. 22a: Nyquist Plots, Blade 3

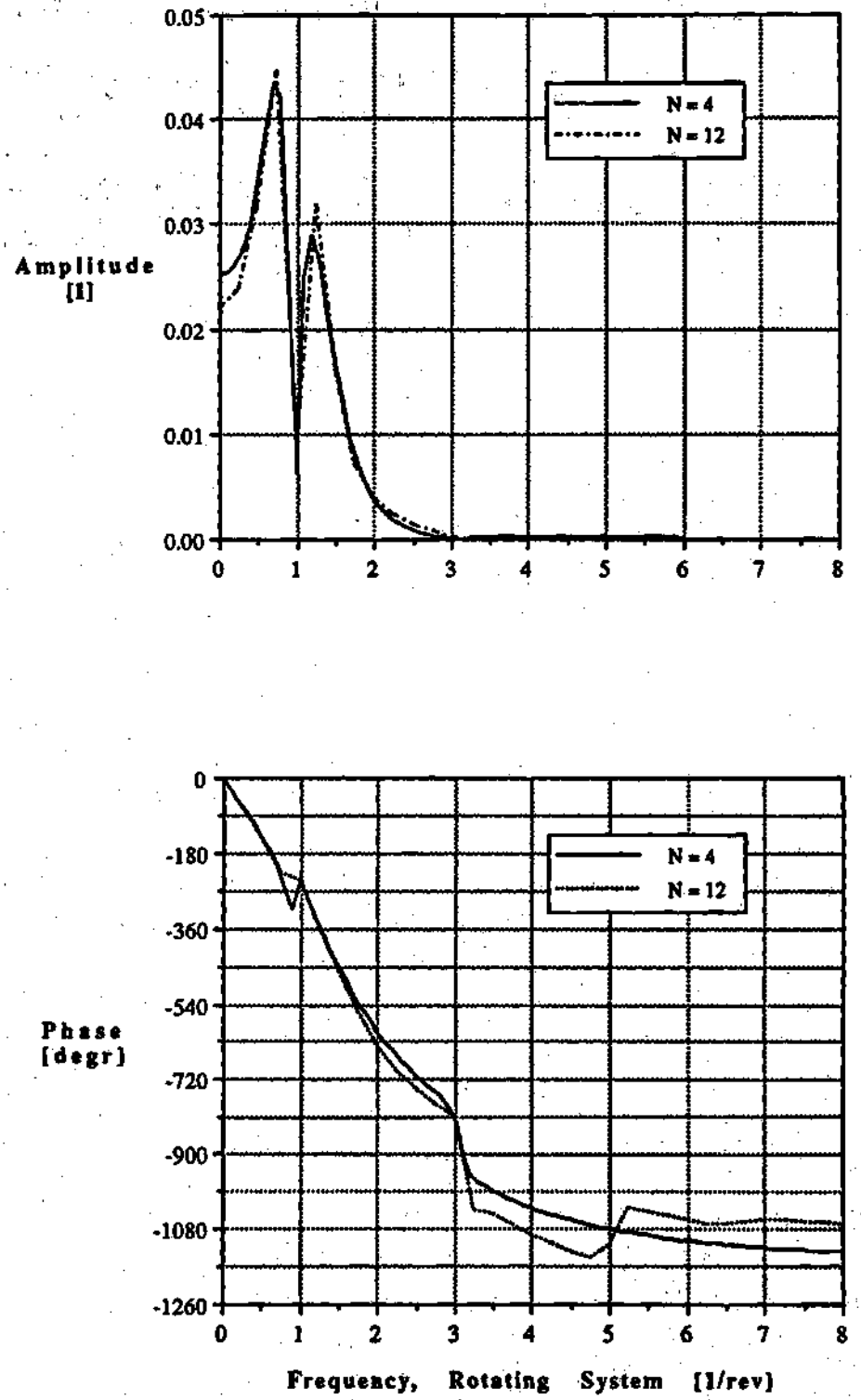


Fig. 22b: Flapping Response, Blade 3

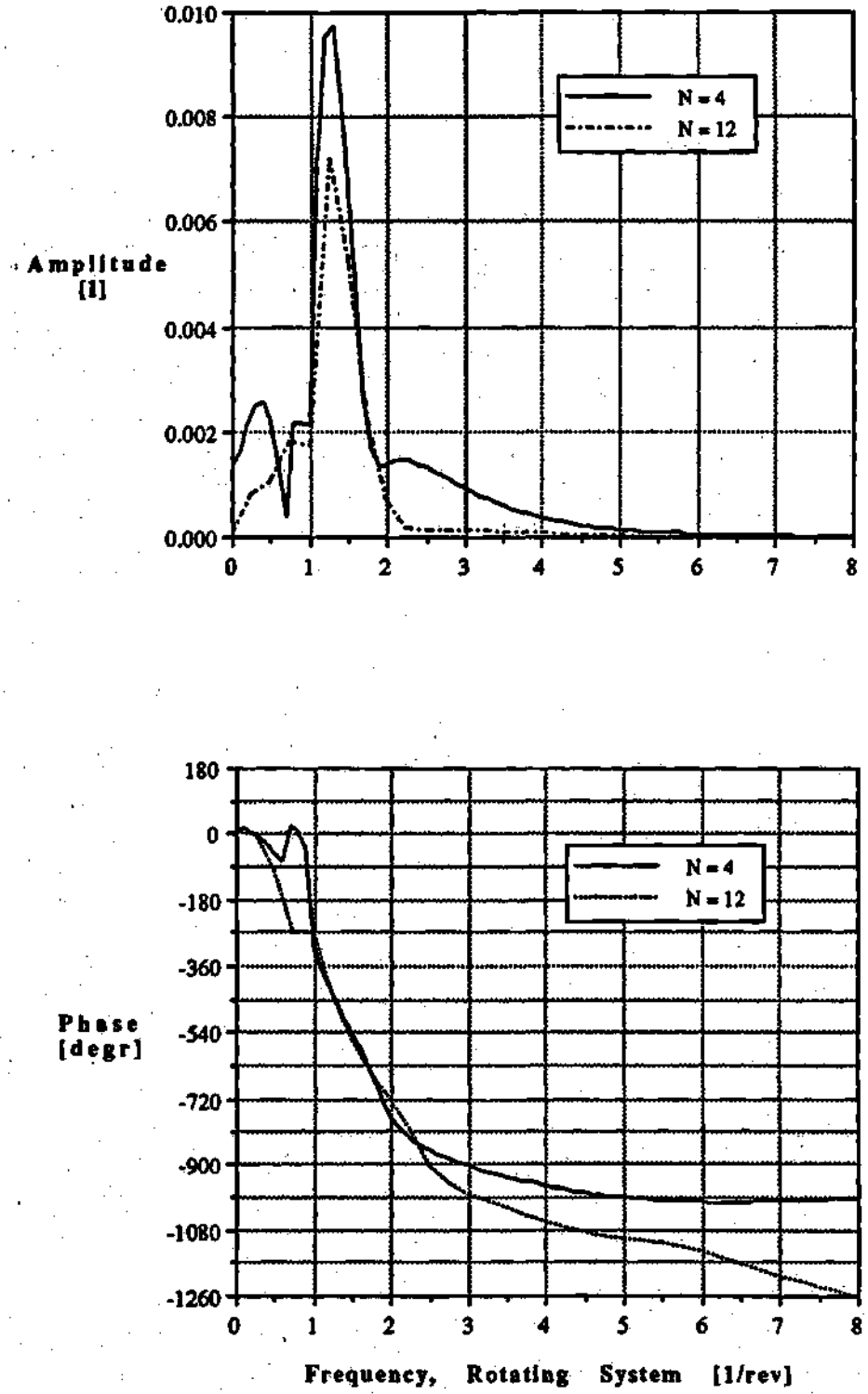


Fig. 22c: Feathering Response, Blade 3

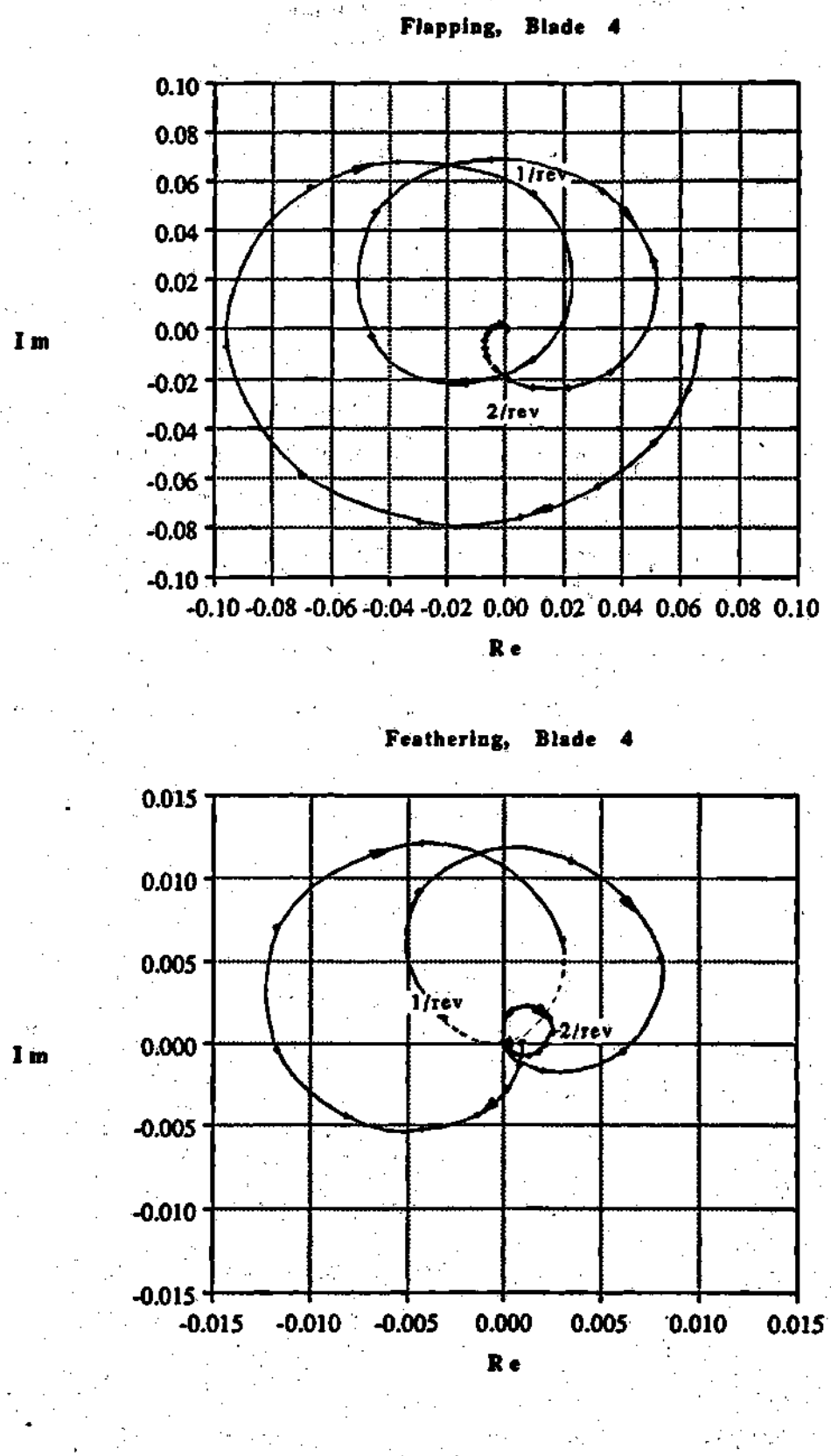


Fig. 23a: Nyquist Plots, Blade 4

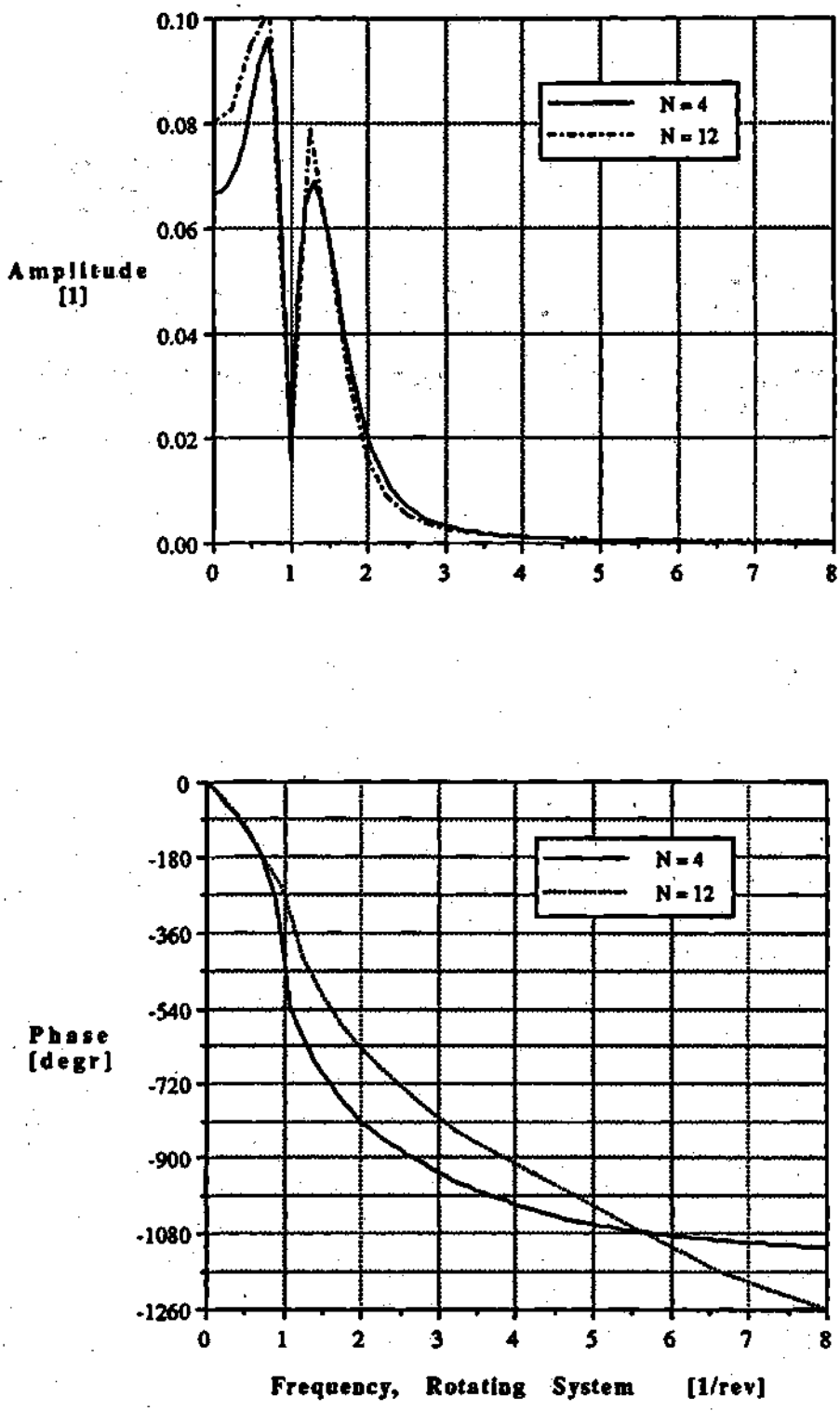


Fig. 23b: Flapping Response, Blade 4

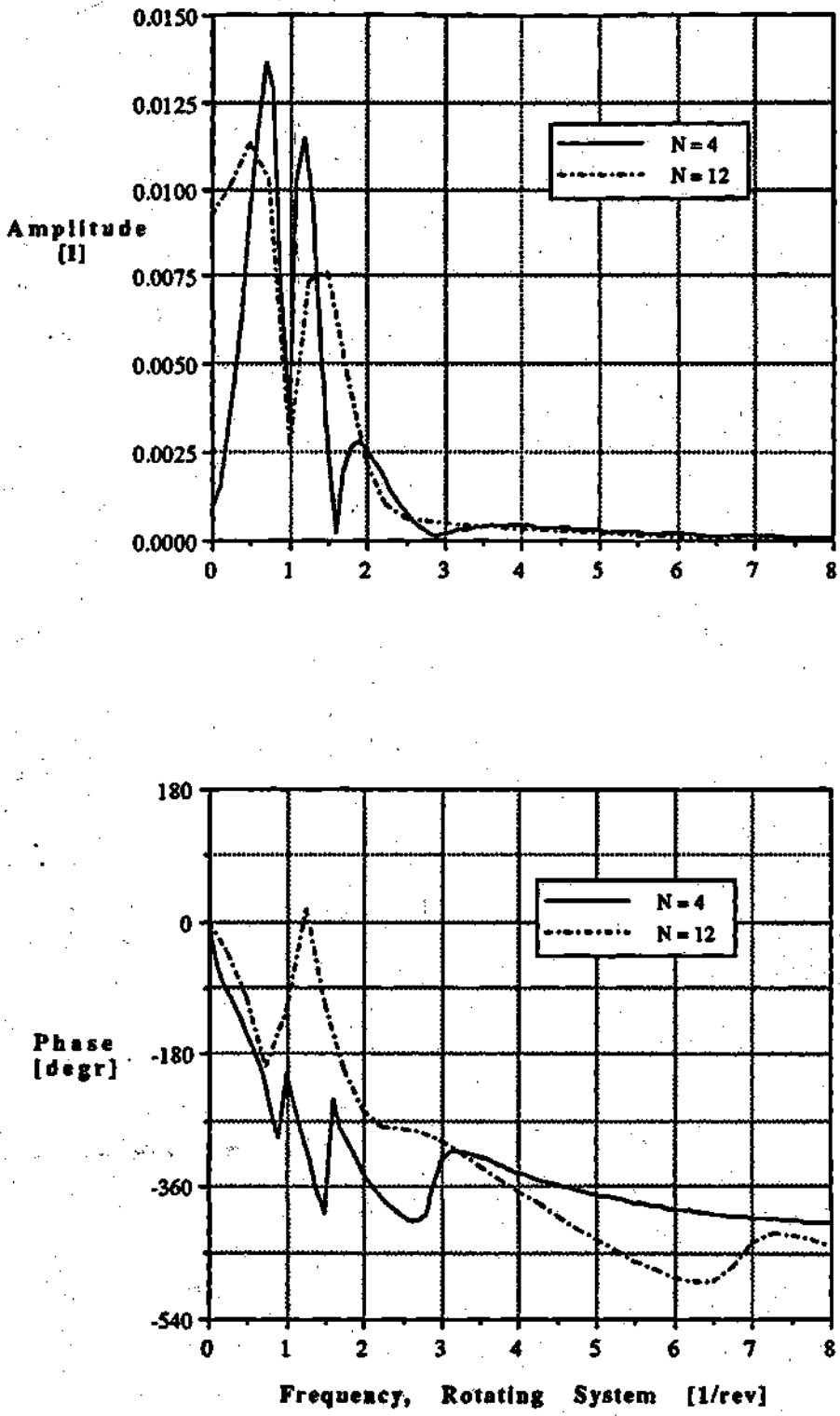


Fig. 23c: Feathering Response, Blade 4

REFERENCES

[1] Miao, W., Kottapalli, S.B.R., and Frye, H.M., "Flight Demonstration of Higher Harmonic Control on S-76," Proceedings of the 42nd Annual National Forum of the American Helicopter Society, Washington, D.C., June 1986.

[2] Walsh, D., "Flight Test of an Open Loop Higher Harmonic Control System on an S-76A Helicopter," Proceedings of the 42nd Annual National Forum of the American Helicopter Society, Washington, D.C., June 1986.

[3] Wei, F.-S., Basile, J.P., and Jones, R., "Vibration Reduction on Servo Flap Controlled Rotor using HHC," presented at the American Helicopter Sociaty Specialists Meeting on Rotorcraft Design, Arlington, Texas, November 1989.

[4] Lemnios, A.Z., and Smith, A.F., "An Analytical Evaluation of the Controllable Twist Rotor Performance and Dynamic Behavior," USAAMRDL Technical Report 72-16, U.S. Army Air Mobility Research and Development Laboratory, Ft. Eustis, Virginia, May 1972, AD 747808.

[5] McLoud, J.L., III, "An Analytical Study of a Multicyclic Controllable Twist Rotor," Proceedings of the 31st Annual National Forum of the American Helicopter Society, Washington, D.C., May 1975.

[6] McCloud, J.L., III, and Weisbrich, A.L., "Wind Tunnel Test Results of a Full-Scale Multicyclic Controllable Twist Rotor," Proceedings of the 34th Annual National Forum of the American Helicpter Society, Washington, D.C., May 1978.

[7] Theodorsen, T., "General Theory of Aerodynamic Instabilities and the Mechanism of Flutter," NASA TR 496, 1949.

[8] Loewy, R.G., "A Two-Dimensional Approach to the Unsteady Aerodynamics of Rotary Wings," *Journal of the Aerospace Sciences*, Vol. 24, No. 2, February 1957, pp. 82-98.

[9] Peters, D.A., and He, C.J., "A Closed-Form Unsteady Aerodynamic Theory for Lifting Rotors in Hover and Forward Flight," Proceedings of the 43rd Annual National Forum of the American Helicopter Society, St. Louis, Missouri, May 1987.

[10] Peters, D.A., and He, C.J., "Comparison of Measured Induced Velocities with Results from a Closed-Form Finite-State Wake Model in Forward Flight", Proceedings of the 45th Annual National Forum of the American Helicopter Society, Boston, Massachusetts, May 1989.

[11] He, C.J., "Development and Application of a Generalized Dynamic Wake Theory for Lifting Rotors," PhD Thesis, Georgia Institute of Technology, July 1989.

[12] Su, A., "Application of a State-Space Wake Model to Elastic Blade Flapping in Hover," PhD Thesis, Georgia Institute of Technology, August 1989.

[13] Wei, F.-S., and Jones, R., "Correlation and Analysis for the SH-2F 101 Rotor," Proceedings of the 28th AIAA/ASME/ASCE/AHS Structures, Structural Dynamics and Materials Conference, Monterey, California, April 1987.

[14] Fitzpatrick, J.E., Collins, R.E., and Lemnios, A.Z., "Technical Description Data for the Kaman 101 Rotor Blade on the SH-2F Helicopter," Report No. R-1815, Kaman Aerospace Corporation, October 1985.

[15] Stevens, W.A., Goradia, S.H., and Braden, J.A., "Mathematical Model for Two-Dimensional, Multi-Component Airfoils in Viscous Flow," NASA-CR 1843

[16] Betz, A., "Höchstauftrieb von Flügeln an umlaufenden Rädern," Zeitschrift der Flugwissenschaften, Vol. 9 (4/5), 1961.

[17] Johnson, W., Helicopter Theory, Princeton University Press, 1980, Chapter 10-7, pp. 526-535.